Computer-based Adaptive Speed Tests

Daniel Bengs[†]

[†]German Inst. for Intern. Educational Research
Information Center for Education
Frankfurt am Main, Germany
bengs@dipf.de

Ulf Brefeld[†‡]

[‡]Technische Universität Darmstadt
Department of Computer Science
Darmstadt, Germany
brefeld@kma.informatik.tu-darmstadt.de

ABSTRACT

The assessment of a person's traits is a fundamental problem in human sciences. Compared to traditional paper & pencil tests, computer based assessments not only facilitate data acquisition and processing but also allow for adaptive and personalized tests so that competency levels are assessed with fewer items. We focus on speeded tests and propose a mathematically sound framework in which latent competency skills are represented by belief distributions on compact intervals. Our algorithm updates belief based on directional feedback; adaptation rate and difficulty of the task at hand can be controlled by user-defined parameters. We provide a rigorous theoretical analysis of our approach and report on empirical results on simulated and real world data, including concentration tests and the assessment of reading skills.

1. INTRODUCTION

The assessment of a person's traits such as ability is a fundamental problem in the human sciences. Perhaps the most prominent examples are the triennial PISA studies launched by the OECD in 1997. Traditionally, assessments have been conducted with printed forms that had to be filled in by the testees, so called paper & 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 [2]. 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 given unlimited time. By contrast, speeded tests deploy homogeneous items that are easy to solve, and testees are discriminated by the time needed to solve the items. In this paper, we focus on pure speed tests akin to [4] as well as tests where response times are assessed together with item correctness, e.g. to study the efficiency of cognitive processes [5].

We devise an algorithm using a data-driven approach for steering the time limits of individual items actively. Items of constant inherent difficulty are administered in a sequence $t = 1, 2, \ldots$, and a limit on response time $\hat{\tau}_t$ is adapted based on testee performance. After the administration of each item, the algorithm chooses the limit for the upcoming item such that as much information as possible on testee's expected response time is collected. The uncertainty of an estimate $\hat{\tau}$ is represented by a belief distribution over a finite interval of admissible response times. When administering item t, an estimate $\hat{\tau}_t$ is drawn, such that $\hat{\tau}_t$ divides the belief mass in two parts whose areas have a predefined ratio roughly corresponding to the odds that the testee responds within the time limit. After the testee attempts solving the item under the time limit $\hat{\tau}_t$, the algorithm receives feedback ϕ_t encoding three cases: (i) if $\hat{\tau}_t - \tau_t < \epsilon$, the time limit $\hat{\tau}_t$ was insufficient for the testee to answer in time and $\phi_t = 1$, (ii) in case $\hat{\tau}_t - \tau_t > \epsilon$, the setting was more than sufficiently long, and $\phi_t = -1$, and (iii) $\tau_t \in [\hat{\tau}_t - \epsilon, \hat{\tau}_t + \epsilon]$ which corresponds to a just right setting and $\phi_t = 0$.

Our learning algorithm is around the following strategy: Once we observe that $\hat{\tau}$ is too small to allow for solving the item, it is highly probable that all time limits $\tilde{\tau} > \hat{\tau}$ would also be too small, and belief in their correctness can be updated. A similar argument holds vice versa for time limits more than sufficiently long. The feedback is therefore used as a directional signal that triggers the update process. In this paper we develop the mathematical framework for computer-based adaptive speed tests and devise an efficient algorithm. We provide a theoretical analysis and report on empirical results using artificial and real-world data.

2. RELATED WORK

Missura & Gärtner [3] consider the problem of dynamic difficulty adjustment as a game between a master and a player that is played in rounds $t = 1, 2, \cdots$, where the master predicts the difficulty setting for the next round based on the player feedback. The authors introduce an algorithm that represents the set of admissible difficulty settings as a finite discrete set $\mathcal{K}$ endowed with a partial ordering. For each of the difficulty levels $k \in \mathcal{K}$, the algorithm maintains a positive number representing belief in k being *just right*. At each round, the prediction allows to update the maximal amount of belief after feedback has been received. In contrast to [3], we use a continuous framework and do not rely on a predefined discrete set admissible settings, but instead find appropriate settings adaptively on the fly.

Csáji and Weyer [1] investigate the problem of estimating a constant based on noisy measurements of a binary sensor with adjustable threshold. That is, of a constant $\theta^* \in \mathbb{R}$ disturbed by additive, i.i.d. noise N_t, only measurements indicating whether the $\theta^* + N_t$ exceeds an adjustable threshold θ_t are available for $t = 1, 2, \cdots$. Under mild assumptions on the distribution of N_t, a strongly consistent estimator is derived, i.e. a method for choosing the thresholds θ_t such that $\theta_t \to \theta^*$ almost surely for any starting value θ_0. In contrast to [1], 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 & Moosbrugger [4] propose the Frankfurt Adaptive Concentration Test II (FACT-II), which conceptualizes concentration as the ability to respond to stimuli in the presence of distractors. After administration of item t, exposure time of the item $t+1$ 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 correct response is given in time or not. Tests using both accuracy and response times are used to assess efficiency of cognitive processes for instance in the measurement of components of reading abilities [5].

3. CAT-FRAMEWORK FOR SPEED TESTS

We consider a computerised adaptive test where a sequence of items of homogeneous difficulty is presented to the testee and response times are recorded. This scenario encompasses adaptive speeded tests such as FACT2 [4] as well as tests targetting efficiency of congitive processes (e.g. [5]). In the former, response times are limited by an adaptation mechanism and relate directly to the trait being assessed. In the latter case, response times are merely observed and used to analyse testee efficiency. Here, testees might take a long time to think and then score perfectly, leading to undesirable ceiling effects, as observed by [5]. Imposing a time limit may increase testing efficiency and also increase variation in item correctness, leading to a higher data quality. We additionally show that our algorithm can be configured to realize a user-defined probability for a timely response.

3.1 Methodology

We consider admissable response times in an interval $T = [a, b]$. We assume that at each position of the testing sequence, there exists a lower bound on the testee's response time which we consider the just right setting $\tau_t \in T$. This is the minimum sustainable response time enabling the testee to solve the item; we assume that it relates to an underlying trait but is independent of the actual item, as the item bank consists of items of constant difficulty. The goal of the adaptation is to iteratively adjust the time limit until the just right setting is reached. To this end, the algorithm maintains a belief distribution $B_t : [a, b] \to (0, \infty)$ on T that is used for accumulating knowledge about the correctness of previously estimated time limits. Correctness of the predictions is assessed after administering each item by feedback ϕ_t, which is based on the relation of the testee's response time τ_t and the predicted time limit $\hat{\tau}_t$: We have $\phi_t = -1$ if $\hat{\tau}_t < \tau_t$, that is, the item is solved within the time limit, $\phi_t = 1$ if the testee runs out of time ($\hat{\tau}_t > \tau_t$), and $\phi_t = 0$ if the item is solved (exactly) at the time limit ($\hat{\tau}_t = \tau_t \pm \epsilon$).

Here, $\epsilon > 0$ is used to decide whether the τ_t is close enough to $\hat{\tau}_t$ to consider $\hat{\tau}_t$ a correct prediction. This is necessary because response times underly random fluctuations and thus the just right time limit remains hidden to the algorithm. Adaptation and prediction is done using the belief function and two preassigned parameters $\beta \in (0, 1)$ and $\delta \in (0, 1)$ as follows: Belief is initialized to be a strictly positive constant on T.[*] The time limit for administering item t is computed as the value $\hat{\tau}_t$ that splits the area under B_t in two parts $P_t(\hat{\tau}_t) := \int_a^{\hat{\tau}_t} B_t(x)dx$ and $Q_t(\hat{\tau}_t) := \int_{\hat{\tau}_t}^b B_t(x)dx$, such that $P_t : Q_t = \delta : 1 - \delta$. Assuming normalized belief, this can be achieved by determining $\hat{\tau}_t$ that satisfies $P_t = \delta$.

It is easy to see that B_t being non-negative by assumption, the mapping $\hat{\tau}_t \mapsto P_t(\hat{\tau}_t)$ is strictly increasing and thus bijective, so $\hat{\tau}_t$ is uniquely determined if only $\int_a^b B_t(x)dx \neq 0$, which because as $B_1 \neq 0$ and $\beta \neq 0$, all $B_t \neq 0$ due to the updating formula given below. After the testee attempts to solve the item given time limit $\hat{\tau}_t$, the algorithm receives feedback ϕ_t indicating whether the time limit was (i) *too long* and $\phi_t = -1$, or (ii) *too short* and $\phi_t = +1$. Because of transitivity, the algorithm may infer that (i) the time was more than sufficiently long or (ii) any shorter time limit would also have been insufficient for the testee. If the testee responded ϵ-close to the time limit and $\phi_t = 0$, no update is necessary because current belief produced a correct prediction. Otherwise, the belief in all settings (i) longer or (ii) shorter, respectively, is lowered by the updating step, which is carried out by multiplying the respective belief values by the learning rate β:

$$B_{t+1}(x) = \begin{cases} \beta B_t(x), & \text{if (i) and } x \geq \hat{\tau}_T \text{ or (ii) and } x \leq \hat{\tau}_t \\ B_t(x), & \text{else.} \end{cases}$$

The parameter β thus controls how much weight is given to information from the current observation; the closer β is to zero, the faster the adaptation. If β is close to 1, the predictions will show less variation. Thus assumptions on the rate of change of the true time limit and the length of the item sequence can be used to guide the choice of β. We give a theoretical analysis yielding bounds on the difference of successive predictions by our algorithm in Theorem 1.

3.2 Computational Aspects

Each feedback step leads to the updating of either the interval $[a, \hat{\tau}_t]$ or the interval $[\hat{\tau}_t, b]$ by multiplying the values of B_t by β. Consequently, for all t, the function B_t belongs to the space of non-negative step functions on $[a, b]$. This allows for efficient storage, manipulation and prediction based on an interval subdivision scheme. Starting with $T = [a, b]$, we divide the interval containing the current prediction $\hat{\tau}_t$ at $\hat{\tau}_t$ and update the belief values to the left or right of $\hat{\tau}_t$ depending on the feedback ϕ_t by multiplying with $\beta \in (0, 1)$. Formally, we write B_t as a sum

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

[*] The initial belief function B_1 can also be tailored to incorporate prior knowledge about where to expect τ_1.

for some $N \in \mathbb{N}$, where $y_i^{(t)} \geq 0$ is the value B_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 $I_{N_t}^{(t)} = [x_{N_t-1}, x_{N_t}]$. The interval endpoints are defined by a partition

$$a = x_0^{(t)} < x_1^{(t)} < x_2^{(t)} < \cdots < x_{N_t}^{(t)} = b$$

of $[a, b]$. By i_t^* we denote the index of the interval containing $\hat{\tau}_t$. If $\phi_t = 1$, we set

$$\begin{aligned} B_{t+1} = & \sum_{i=1}^{i_t^*-1} \beta y_i \chi_{I_i^{(t)}} + \beta y_{i_t^*} \chi_{[x_{i_t^*-1}, \hat{\tau}_t)} \\ & + y_{i_t^*} \chi_{[\hat{\tau}_t, x_{i_t^*})} + \sum_{i=i_t^*+1}^{N_t} y_i \chi_{I_i^{(t)}}, \end{aligned} \tag{1}$$

if $\phi_t = -1$, belief at $t+1$ is defined analogously, that is, nodes $x_\cdot^{(t+1)}$ are as above, but the weights with indexes greater or equal than i_t^* are multiplied by β. Finally, if $\phi_t = 0$ no update is necessary and $B_{t+1} = B_t$. The belief function can be stored and updated efficiently by storing the endpoints $x_1^{(t)}, \cdots, x_{N_t-1}^{(t)}$ and function values $y_1, \cdots, y_N$. Theorem 1 bounds the minimal and maximal difference between successive estimates of the algorithm.

THEOREM 1. *Let* $(\hat{\tau}_t)_{t=1}^N$ *be a sequence of estimations of the CAST algorithm with parameters* β *and* δ. *Then for* $t = 1, \cdots, N-1$ *it holds that*

$$\frac{\delta(1-\delta)(1-\beta)}{\max_{x\in[a,b]} B_t(x)} B \leq |\hat{\tau}_{t+1} - \hat{\tau}_t| \leq \frac{\delta(1-\delta)(1-\beta)}{\min_{x\in[a,b]} B_t(x)} B,$$

where $B = \int_a^b B_t(x)dx$.

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, then 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. The dependance on δ can be interpreted as follows: The more δ deviates from 0.5, the more will inital time limits be biased towards a or b resp. and also adaptation to the observed time limit will be slower. Therefore, δ can be regarded a parameter controlling difficulty bias. Our experiments demonstrate that by varying δ, a wide range of difficulty settings can be realized. We verify this claim in the next Section.

4. EMPIRICAL RESULTS

4.1 Artificial Data

To showcase the adaptivity of our approach, 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 CAST to state-of-the-art baselines POSM [3], Csáji-Weyer-Iteration (CWI) [1], and the algorithm used by FACT-II [4].

Throughout this suite of experiments, we use $T = [0, 1]$. To allow for a fair comparison, the set of difficulty settings for POSM consists of N equidistantly sampled points in T, where n is the number of time steps used. This choice guarantees that the number of subdivisions made by CAST is less than or equal to the number of settings available to POSM.

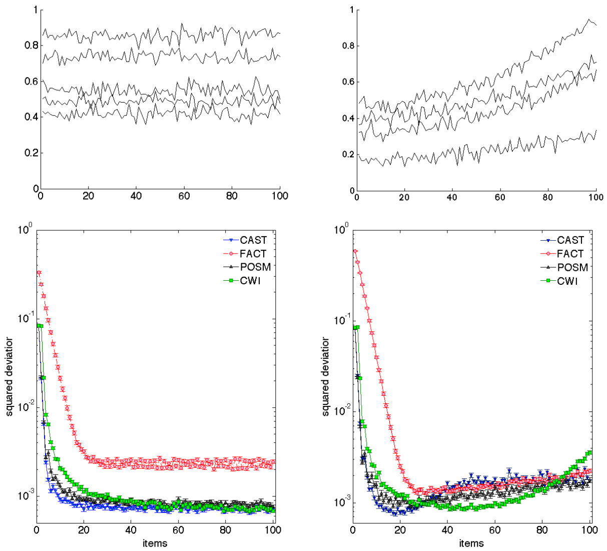

Figure 1: Top: Artificial response times. Bottom: Results for constant (left) and drift (right) scenarios.

Thus, all approaches have access to the same amount of resources. We use optimal parameters for CAST and POSM chosen by model selection.

We study the behavior of the algorithms in constant and dynamic scenarios: In the first setting, the ground-truth τ remains constant. We sample the constants from a uniform distribution on T. In the second setting, we simulate learning and tiredness effects of testees. The true parameter τ thus underlies drifts and the resulting distribution is not stationary. In both settings, simulated response times are additionally disturbed by white noise. Figure 1 (top row) shows sample observations for the two scenarios. We report on average deviations of 1,000 repetitions with randomly generated τ.

Figure 1 (bottom, left) shows the results for the constant setting. All algorithms need some time to adapt to the noisy τ with FACT showing severe problems in the estimation process and finally oscillating between two estimations that are both far away from the simulated ground-truth. The best adaptation is achieved by CAST in terms of speed as well as overall performance. CWI converges to a comparable estimator at the end of the sequence but the adaptation process is not as fast. POSM performs only slightly worse than CAST. The visual differences are reflected in Table 1 that summarizes the results.

Figure 1 (bottom, right) shows the results for the dynamic scenario containing drift. Again FACT is significantly outperformed by the competitors. CWI describes a U-shaped curve and proves not appropriate for dynamic scenarios due to the strict assumptions on the data generating distribution. By contrast, CAST converges quickly to the initial plateau after about 20 responses and looses accuracy when the drift begins to dominate the scenario. POSM takes again more time to adapt to the data but shows a slightly improved performance for intermediate items which also leads to the smallest difference in Table 1. However, note that

Table 1: Sum of squared deviations of Figure 1

	CAST	POSM	FACT	CWI
constant	1.8752	2.0427	14.5896	2.9801
drift	2.6661	2.4396	24.5116	3.4407

we tailored the discrete POSM to the continuous scenario to obtain a fair comparison in terms of computational resources. In real world settings, the optimal discretization of POSM is not obvious and often intractable. CAST can thus be seen as the best off-the-shelf approach although POSM achieved slightly better scores in the dynamic scenario.

4.2 Reading Skills

In this section we evaluate our algorithm in an experiment using real world data from a computerized test of phonological representation by Richter et al. [5]. Testees listen to an auditorial reference stimulus in form of a pseudo word. The presentation is immediately followed by a displayed pseudo word on the screen. The testee's task is to decide whether the displayed word is phonologically identical to the auditory one. No time limit is enforced.

The data consists of response times of 528 children, between five and 11 years old, assessed during a test comprising of $n = 64$ items. We simulate the effects of incorporating a time limit by our algorithm as follows: After preprocessing by removing extreme response times (>2500ms) and compensating the strong linear relationship between number of syllables and mean response time ($R^2 = .83$) to level item difficulty, the linearly transformed response times are between -932.34 and 2267 ms. We use our algorithm to predict expected response time for each participant on the interval $[-1000, 2500]$.

Note that without time limits, ceiling effects in accuracy may be observed [5] while too tight limits on response time can easily lead to frustrated participants. We focus on the proportion of items each participant would not have answered in time for different values of β and δ. The goal is to predict for each participant time limits on each item, such that a non-zero chance of solving the respective item is realized. We compute predictions $\hat{\tau}_i; i = 1, \cdots, 64$ for each participant and analyse the proportion P of items not solved within the predicted time limit and compare the results with the proportion achieved by using percentiles of the testee's response times. We use $\epsilon = 10$ms.

Figure 2 (top, left) shows the distributions of P across participants for $\beta = 0.65$ and $0.05 \leq \delta \leq 0.95$. The figure indicates that proportions P between 20% and 65% can be robustly realized by using different values of difficulty bias δ. By contrast, the proportions realized by a percentile-based approach in Figure 2 (top, right) span a broader range but contain much variance across the participants, showing that our adaptive approach leads to a more homogeneous experience across testees. For our algorithm, dispersion measured by range is at roughly 20 percentage points across all δ while for percentiles, ranges between 20 and 70 percentage points are observed.

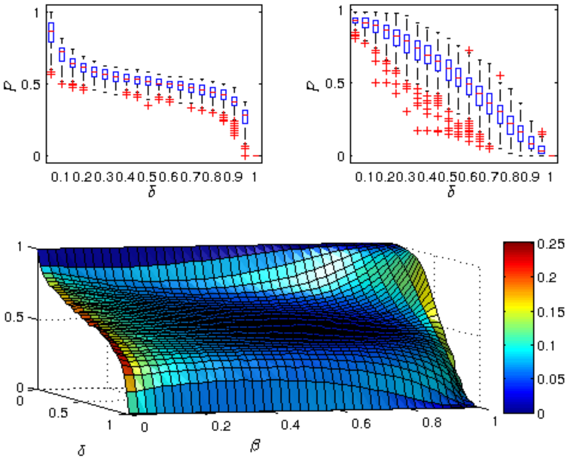

Figure 2: Results for the reading skill experiment.

Figure 2 (bottom) shows mean proportions P of testees not responding in time on the z-axis while the color corresponds to the standard deviation at every point. The figure shows that a low dispersion and a wide range of proportions can be set with our algorithm also when the β parameter is varied; mean proportions are stable for all but extreme values of both parameters. In sum, our algorithm effectively controls the adaptation in both difficulty bias and adaptation rate.

5. CONCLUSION

We introduced a novel technique for computer-based adaptive speed tests. In contrast to existing methods, our approach is devised from a mathematically sound framework and maintains belief distributions on compact intervals to represent estimates of the unknown parameter. In addition, our approach is purely data-driven and does not rely on assumptions on the distribution of the true parameter. Empirically, we showed the effectiveness of our adaptive speed test on artificial and real world scenarios.

Acknowledgements

We are grateful to Johannes Naumann and Yvonne Neeb for sharing the data of the reading skill experiment with us.

6. REFERENCES

[1] Csáji, B.C., Weyer, E.: System identification with binary observations by stochastic approximation and active learning. IEEE CDC-ECE, pp. 3634–3639 (2011)

[2] Furr, R.M., Bacharach, V.R.: Psychometrics: an introduction. SAGE Publications, Incorporated (2007)

[3] Missura, O., Gärtner, T.: Predicting Dynamic Difficulty. Advances in Neural Information Processing Systems 24, pp. 2007–2015 (2011)

[4] Moosbrugger, H., Goldhammer, F.: FAKT-II Frankfurter Adaptiver Konzentrationsleistungs-Test II. Huber, Bern (2007)

[5] Richter, T., Isberner, M.B., Naumann, J., Kutzner, Y.: Prozessbezogene Diagnostik von Lesefähigkeiten bei Grundschulkindern. Zeitschrift für Pädagogische Psychologie 26(4), 313–331 (2012)