

令和2年度 情報数理工学プログラム卒業論文概要

平成 29 年度 入学	学籍番号 1710003
指導教員 植野 真臣	氏名 赤坂 尚紀
題目	決定木を用いた適応型テストの多階層木圧縮による生成時間削減

概要

本論文では、決定木を用いた適応型テストのための多階層木圧縮による生成時間削減手法を提案する。適応型テストとは、コンピュータ・テストイング手法の一つで、受験者の回答のたびに能力を逐次的に推定し、その推定能力値に対して情報量が最も高い項目を出題する手法である。受験者の能力に応じて項目を出題することで、受験者の測定精度を減少させることなく、出題項目数や受験時間を軽減することができる。項目選択に出題分野や項目露出など様々な制約を課した適応型テストが提案されているが、それらの制約付き手法は項目選択の計算に時間を要する。この項目選択時間を削減するために、Uenoら(2010)は事前に全ての受検者の回答パターンに対する項目決定木を生成する手法を提案した。この手法はテスト中の項目選択時間を軽減し、受験者に待ち時間なく項目を出題できることから、近年この手法を基にした適応型テストが数多く提案されている。その一つとして、Rodríguez-Cuadrado(2020)らは決定木を用いた適応型テストの課題である、分枝数の指数的增加に伴う時間・空間計算量の問題を緩和するために、決定木の同一階層の分枝のうち推定能力値とその分布が類似するものをマージする Merged Tree-CAT という手法を提案した。この手法は分枝数の増加を軽減し決定木生成に要する時間を削減したが、同一階層のみしかマージできないため出題項目数やアイテムバンクの大きさによっては依然として決定木の生成に時間を要する。本論文では、決定木生成時間の削減を目的として、Merged Tree-CAT における同一階層の分枝のマージに加え、既に生成した上位の階層の頂点へのマージも行う手法を提案し、本手法が能力推定精度を維持しつつ、決定木生成時間を削減できることをシミュレーション及び実データを用いて示した。

決定木を用いた適応型テストの多階層木圧縮による生成時間削減

2021年2月26日

情報数理工学プログラム

学籍番号 1710003

赤坂 尚紀

指導教員 植野 真臣

決定木を用いた適応型テストの多階層木圧縮による 生成時間削減

植野研究室
1710003 赤坂 尚紀

2021 年 2 月 26 日

目次

1	まえがき	2
2	項目反応理論	3
3	適応型テスト	4
4	決定木を用いた適応型テスト	5
4.1	DT	6
4.2	Tree-CAT	7
4.3	Merged Tree-CAT	9
5	提案手法	11
6	評価実験	16
6.1	実験 1:従来手法との比較	16
6.2	実験 2: Merged Tree-CAT との比較	19
6.3	実験 3:圧縮率に関する要因分析	21
7	むすび	25

1 まえがき

近年、e テスティングの実用化が注目されている。e テスティングは一回のテストの推定精度を高めるだけでなく、異なるテストを何度受験しても同一尺度上で受験者の能力を評価できるという利点があることから、テストの結果が受験者に大きな影響を及ぼす資格試験や入学試験を含む様々なテスト場面において導入が進んでいる。

また e テスティングの技術の一つとして、適応型テスト (Computer Adaptive Test: CAT) が知られている。適応型テストでは、受検者の解答のたびに能力を推定し、その能力推定値に対して情報量が最も高い項目を出題する。このように受検者の能力に応じて項目を逐次的に出題することで、受検者の測定精度を減少させることなく、出題項目数や受験時間を軽減できる利点がある。項目選択に出題分野や項目露出など様々な制約を課した CAT 手法が提案されているが、それらの制約付き手法は項目選択の計算に時間を要する。

この項目選択時間を削減するために、Ueno and Songmuang(2010) [2] は、事前にすべての受検者の回答パターンに対する項目決定木を生成する DT という手法を提案した。本手法は予め決定木を生成しておくことで、テスト中の項目選択時間を軽減し、受験者に待ち時間なく項目を出題できることから、近年この手法を基にした適応型テストが数多く提案されている [3-5]。

具体的には、Ueno(2013) [5] らは、Expected Value of Test Information: EVTI という情報量を開発し、DT を用いる手法を提案した。EVTI は、従来の CAT で広く用いられているフィッシャー情報量に比べ、項目選択の偏りが少なく、テスト序盤の推定誤差が小さい等の利点があるが、計算に非常に長い時間を要する。この手法は DT を用いることで受験者の待ち時間を短く保ったまま EVTI を用いた CAT を実現した。

また、項目露出を制限する CAT 手法として、Restricted 法 (Revuelta and Ponsoda, 1998) [20], Randomesque 法 (Kingsbury and Zara, 1989; Shin, 2017) [18], Sympton-Hetter 法 (Sympton and Hetter, 1985) [12], Eligibility 法 (van der Linden, 2003) [19], Shadow test (van der Linden and Veldkamp, 2005) [10] といった様々な手法が提案されているが、これらの手法には受験者の待ち時間が増加してしまう。また、上にこれらの手法のいくつかは、二人以上の受験者に同時にテストを実施することができないという問題がある。このような問題に対し、Delgado-Gomez(2019) et al. [3] らは、露出率を決定木を用いた CAT において項目露出を制限する手法として Tree-CAT を提案した。

Rodríguez-Cuadrado(2020) et al. [4] らは、決定木を利用した CAT の課題である、分枝数の指数的増加に伴う時間・空間計算量の問題を緩和するために、決定木の同一階層の分

枝のうち推定能力値とその分布が類似するものをマージする Merged Tree-CAT という手法を提案した。この手法は、分枝数の増加を軽減し、決定木生成に要する時間を削減したが、同一階層のみしかマージできないために出題項目数やアイテムバンクの大きさによっては決定木の生成に要する時間の問題は依然として存在する。

本論文では、Merged Tree-CAT において同一階層の分枝同士のマージに加え、既に生成した階層の頂点へのマージを行う手法を提案し、能力推定精度を維持しつつ、決定木を圧縮し生成時間を削減させることを示す。本手法の有効性をシミュレーションデータを用いて示した。

2 項目反応理論

本章では、本研究の基礎理論として用いる項目反応理論について述べる。項目反応理論は、コンピュータテストを実行する際に用いるテスト理論の一つであり、以下のような特徴をもつ。

- (1) 測定精度の低い異質項目の影響を少なくして受験者の能力推定が可能。
- (2) 異なる項目への受験者の反応を同一尺度上で評価が可能。
- (2) 欠測データからパラメータの推定が可能。

項目反応理論は適応型テストや等質テストといった現在のコンピュータテスト運用の基礎となる理論であり、情報技術者試験「IT パスポート」や医療用共用試験等の評価場面で広く利用されている。本論文では項目反応理論の中でも最も広く利用されている 2 母数ロジスティックモデル (2PLM:2-Parameter Logistic Model) を用いる。

2 母数ロジスティックモデルは、正誤判定問題や多肢選択問題など、データが正誤の 2 値となる反応データに適用できる項目反応モデルで、古くから広く利用されている。2PLM では能力値 $\theta \in (-\infty, \infty)$ をもつ受験者がテスト項目 $i \in 1, \dots, I$ に正答する確率を以下の式で表す。

$$p(u_i = 1|\theta) = \frac{1}{1 + \exp[-1.7a_i(\theta - b_i)]} \quad (1)$$

ここで、 u_i は受検者が項目 i に正答する場合 1、誤答する場合 0 とする。また、 $a_i \in [0, \infty)$ と $b_i \in (-\infty, \infty)$ はそれぞれ項目 i の識別力パラメータと難易度パラメータである。難易度パラメータ b_i は能力値と等しいとき、すなわち $b_i = \theta$ のときその項目 i への正答確率が 0.5 となり、能力値が $\theta = b_i$ の受検者の能力を精度良く評価することができる。また、識別力パラメータ a_i が高い項目ほど $\theta = b_i$ 付近の受検者の能力を精度良く評価することができる。

3 適応型テスト

項目反応理論を用いた一般的な適応型テストは図 1 のように、受験者の回答データを基に能力を逐次的に推定し、その推定値に対して、アイテムバンクとよばれる出題候補となる項目群から情報量の高い項目を選択し出題する、ということを推定能力値が収束するまで繰り返す。

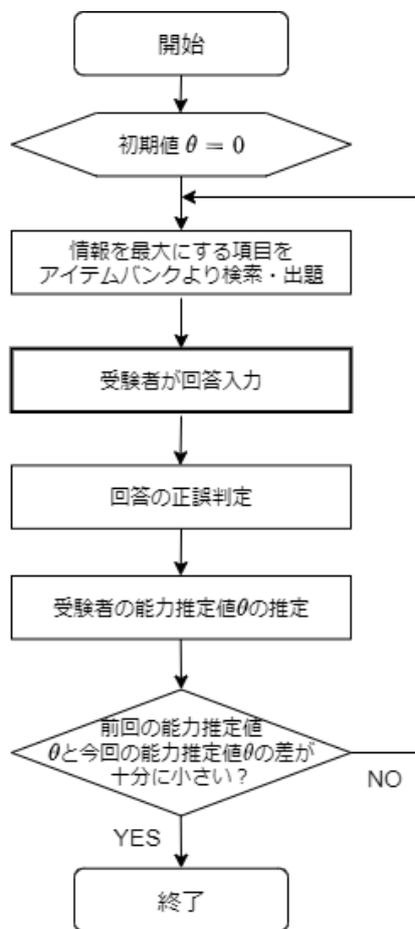


図 1 適応型テストのアルゴリズム

受験者の能力推定には EAP(expected a posteriori) 推定 (Bock & Mislevy, 1988)[1] と、MAP(maximum a posterior) 推定 (Lord 1986, Mislevy 1986)[2] が多く用いられるが、今回は提案手法でも用いている EAP 推定について説明する。受験者の最初から $m - 1$ 個までの出題項目への反応から推定された能力値を $\hat{\theta}_{m-1}$ とし、受験者のそれまでのテストへの反

応履歴を $k_{i_1}, \dots, k_{i_{m-1}}$ とすると, EAP 推定は式のような式 (2) で表される.

$$\hat{\theta}_{m-1} = \int_{-\infty}^{\infty} \theta f(\theta | k_{i_1}, \dots, k_{i_{m-1}}) d\theta \quad (2)$$

ここで $f(\theta | k_{i_1}, \dots, k_{i_{m-1}})$ は推定能力値の事後確率分布を表しており, $f(\theta)$ を推定能力値の事前確率分布, $P_i(\theta, k_i)$ を項目反応理論から得られる能力値 θ の受験者が項目 i に対して回答 k_i を選ぶ確率とすると, ベイズの定理より, 式 (3) で与えられる.

$$f(\theta | k_{i_1}, \dots, k_{i_{m-1}}) = \frac{P_i(\theta, k_i) f(\theta)}{\int_{-\infty}^{\infty} P_i(\theta, k_i) f(\theta) d\theta} \quad (3)$$

項目選択に用いられる情報量は使用される項目選択基準によって異なり, 項目選択基準には FMI(Fisher Maximum Information)(Lord, 1980; Weiss, 1982) [14, 15] や, EPV(Expected Posterior Variance)(van der Linden and Pashley, 2009) [9], MLWI(Maximum Likelihood Weight Information)(Veerkamp and Berger, 1997) [16], KL(Kullback-Leibler information)(Chang and Ying, 1996) [13], MI(mutual information)(Weissman, 2007) [17] 等がある. これらの項目選択基準は一般的に, 数値積分を含まない計算コストの低いものは, テストの序盤における大きな推定誤差や, 項目選択の大きな偏りといった問題を持ち, それらの問題をもたないものは, 高い計算コストをもつという傾向がある. そのため, 高精度で項目選択に偏りのない項目選択基準を用いる場合, 受験者の回答の後, 次の項目を選択, 出題するまでの待ち時間が長くなってしまいう傾向がある.

また上に挙げた項目選択基準では, アイテムバンク内の一部の項目が広範囲の能力値に対して最大の情報量をもつことから, 一部の項目の過剰露出が問題となっている. この問題を解決するために, Randomesque 法 (Kings bury and Zara, 1989; Shin, 2017) [18], Sympson-Hetter 法 (Sympson and Hetter, 1985) [12], Eligibility 法 (van der Linden, 2003) [19], Shadow test(van der Linden and Veldkamp, 2005) [10], Restricted 法 (Revuelta and Ponsoda, 1998) [20] といった項目露出を制御する手法が提案されているが, これらの手法は, すでに問題となっている項目選択にかかる時間をさらに増加させてしまう (Delgado-Gomez, Laria, and Ruiz-Hernandez, 2019) [3]. また, これらの手法の内いくつかは, 一人の受験者にテストを実施するたびに, 各項目の露出率を計算し直す必要があるため, 複数の受験者に対して同時にテストを実施することができないという問題もある.

4 決定木を用いた適応型テスト

本節では提案手法に関連のある手法を紹介する.

4.1 DT

Ueno & Songmuang(2010) [2] によって, すべての受験者の回答パターンに対する項目決定木をテスト実施前に生成する DT という手法が提案されている. この手法で扱う決定木は図 2 のような構造をもつ.

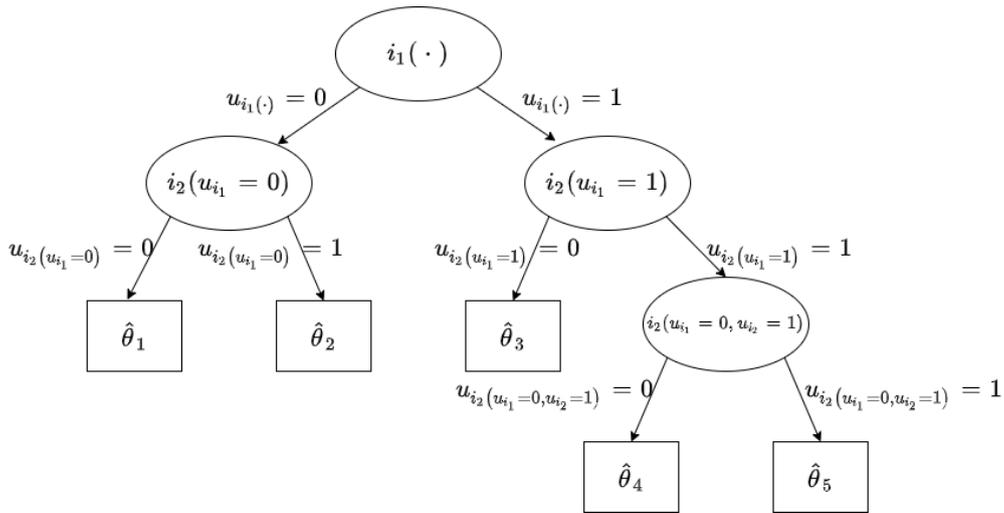


図 2 Ueno & Songmuang(2010) で提案された決定木構造の例

$i_m(u_{i_1}, \dots, u_{i_{m-1}})$ は, $m-1$ 問目までの回答パターンが $u_{i_1}, \dots, u_{i_{m-1}}$ である受験者に m 番目に出題される項目を表し, $u_{i_m}(u_{i_1}, \dots, u_{i_{m-1}})$ は項目 $i_m(u_{i_1}, \dots, u_{i_{m-1}})$ に対する回答を表し, その値は正答であれば 1, 誤答であれば 0 をとる.

この手法は従来の適応型テストがもつ, 受験者が項目に回答してから, 次の出題項目が出題されるまでの待ち時間を軽減することができ, その他にも, 高い推定精度をもつことから能力推定に必要な項目数が少なく済む, 項目選択の偏りを軽減することができるといった利点をもつ.

また, ueno(2013) [5] らは, 従来の適応型テストで一般的に項目選択基準として用いられる FI(Fisher Information) に比べ, 項目の選択において偏りが少なく, テストの序盤における推定誤差が少ないといった利点がある, EVTI(Expected Value of Test Information) という項目選択基準が提案している. EVTI には計算コストが非常に高いという特徴があるが, DT を用いることで EVTI を用いた適応型テストを実現した.

4.2 Tree-CAT

Delgado-Gomez, Laria, and Ruiz-Hernandez(2019) [3] は, 従来の適応型テストがもつアイテムバンク内の一部の項目の過剰露出の問題を改善させるために, Tree-CAT という, 項目の露出度を制御する決定木を用いた適応型テスト手法を提案した. Tree-CAT では, 図 3 のような決定木を生成する.

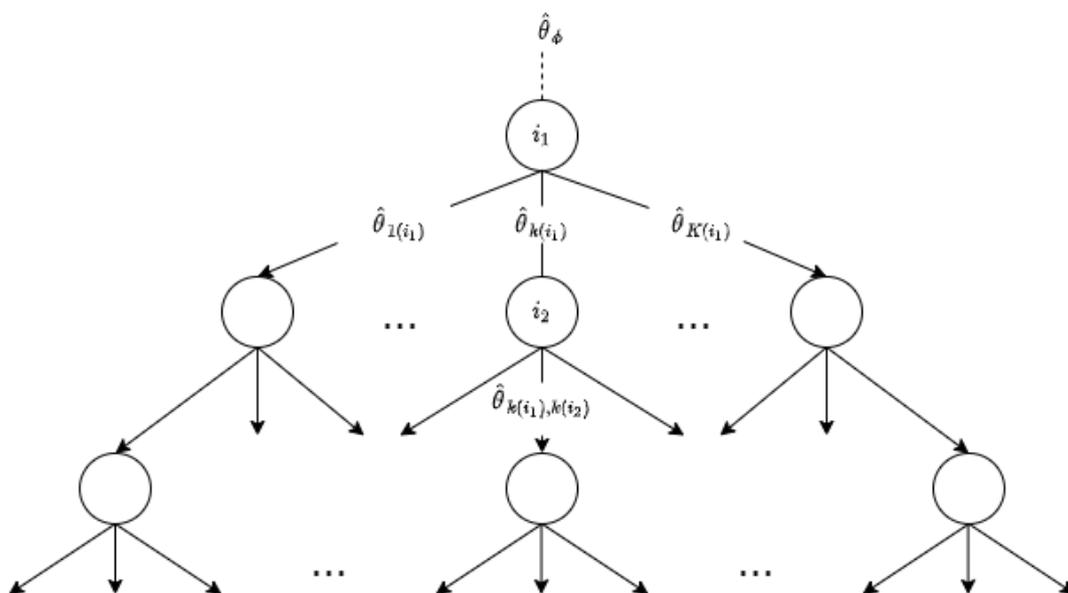


図 3 Tree-CAT で生成される決定木の構造

各頂点は割り当てられた項目 i とその頂点に到達した受験者の推定能力値 $\hat{\theta}$ をもつ. 各分枝は各頂点に割り当てられた項目への回答 k_i とその回答を選んだ場合の事後確率分布に対応する. 事後確率分布は式 (3) のような形で求められ, それを基に, 推定能力値を式 (2) で示した EAP 推定を用いて計算する.

決定木の生成は各分枝に対応する推定能力値を求めるところから始まる. 続いて各分岐先の頂点に対してアイテムバンク内の各項目がもつ情報量を計算する. Tree-CAT では, その項目が出題される前の推定能力値と, 出題された後の各分枝に対応する推定能力値の平均二乗誤差 (MSE: mean squared error) の合計値が小さいものを情報量が高い項目とみなす. Delgado-Gomez, Laria, and Ruiz-Hernandez(2019) [3] で決定木を用いた適応型テストで, 平均二乗誤差を最小化することが, 通常の適応型テストにおいて, EPV(Expected Posterior Variance)(van der Linden and Pashley, 2009) を最小化することに等しいことが示

されている. 各分岐先の頂点に各項目を割り当てた場合平均二乗誤差の合計値 G_i^n を以下式 (4) のように定める.

$$G_i^n = \begin{cases} \int_{-\infty}^{\infty} \left(\sum_{k_i=1}^{K_i} (\theta - \hat{\theta}_n^{k_i})^2 P_i(\theta, k_i) \right) f_n^{k_s}(\theta) d\theta & (i \notin A_u^{k_s}) \\ \infty & (i \in A_u^{k_s}) \end{cases} \quad (4)$$

ここで, $\hat{\theta}_n^{k_i}$ を割り当てられた項目 i に対して回答 k_i を選択した場合の推定能力値, $P_i(\theta, k_i)$ を能力 θ の受験者が項目 i に対して回答 k_i を選ぶ確率, $f_n^{k_s}(\theta)$ を頂点 n に繋がる分枝と対応する推定能力値の確率分布, $A_u^{k_s}$ を頂点 n に到達した受験者が今までに回答した項目集合とする.

$A_u^{k_s}$ に含まれる項目の G_i^n を非常に大きな値を設定しておくことで, 一つのテストにおいて同じ問題が複数回出題されないようにする.

各項目が各分岐先の頂点に対してもつ情報量を求めた後, 以下の線形最適化問題を解くことで, 各項目の露出率を制限しながら, 各頂点に割り当てる項目を選択する.

$$\begin{aligned} & \min \sum_{i=1}^N \sum_{n=1}^{Z_m} \alpha_i^n G_i^n \\ & \text{s.t.} \\ & \sum_{i=1}^N \alpha_i^n = D_u^{k_s} \quad n = 1, \dots, Z_m \\ & \sum_{n=1}^{Z_m} \alpha_i^n \leq c_i^m \quad i = 1, \dots, N \\ & \alpha_i^n \geq 0 \quad i = 1, \dots, N \quad n = 1, \dots, Z_m \end{aligned}$$

ここで, α_i^n を頂点 n に項目 i が割り当てられる確率, $D_u^{k_s}$ を受験者が頂点 n の親頂点 u に到達し項目 s に対し回答 k_s を選ぶ確率, c_i^m を階層 m まで決定木を生成した後の各項目の利用可能率とする. この線形最適化問題は, あらかじめ設定した各項目の利用可能率を超えないように制約をかけながら, 各頂点に対して現在の推定能力値と項目に回答した後の推定能力値の平均二乗誤差ができるだけ小さくなるような項目を割り当てることを目的としている.

各頂点への項目の割り当てが完了した後, 割り当てた項目の回答パターンの数だけ分枝が発生し, 再び各分枝に対応する推定能力値を計算する. 以上のような工程を決定木の階層が決められた値 M となるまで繰り返すことで決定木が生成される.

このような決定木を事前に生成することで, 項目露出を制御しながら, 受験者の待ち時間

もなく、高精度な適応型テストを実施可能になった。この手法は、従来の適応型テストが抱える、項目選択に伴う受験者の待ち時間の問題と、一部の項目の過剰露出の問題を改善した。しかしその一方で、従来からの決定木を用いた適応型テストの問題点である、分枝数の指数的増加に伴う時間・空間計算量の爆発的増加に関しては、依然として問題がある。

4.3 Merged Tree-CAT

Tree-CAT が抱えている、分枝数の指数的増加に伴う時間・空間計算量の問題を軽減することを目的として、Rodríguez-Cuadrado, Delgado-Gómez, and Laria(2020) [4] によって Merged Tree-CAT という手法が提案されている。Merged Tree-CAT では、Tree-CAT において各分岐先の頂点に割り当てる項目選択をする前に、同一階層の分枝のうち、対応する推定能力値とその分布が類似しているものをマージすることで枝刈りを行い、決定木の肥大化を抑制する。

具体的には、以下の条件式 (5)(6) または (6)(7) を満たす分枝のペアを推定能力値とその分布が類似しているものとして、図 4 のような形でマージする。

ここで、 Z_{m-1} を階層 $m-1$ に存在する頂点数、 K_{i_n} を階層 $m-1$ 内の頂点 n に割り当てられた項目 i に対する回答パターン数、 $\hat{\theta}_n^{k_i}$ を階層 $m-1$ 内の頂点 n に割り当てられた項目 i に対して回答 k_i を選んだ場合の受験者の推定能力値、 L_1 と L_2 を事前分布の上限と下限、 $f_n^{k_i}(\theta)$ を階層 $m-1$ 内の頂点 n に割り当てられた項目 i に対して回答 k_i を選んだ場合の受験者の推定能力値の事後確率分布、 δ をあらかじめ設定した最小類似度とする。

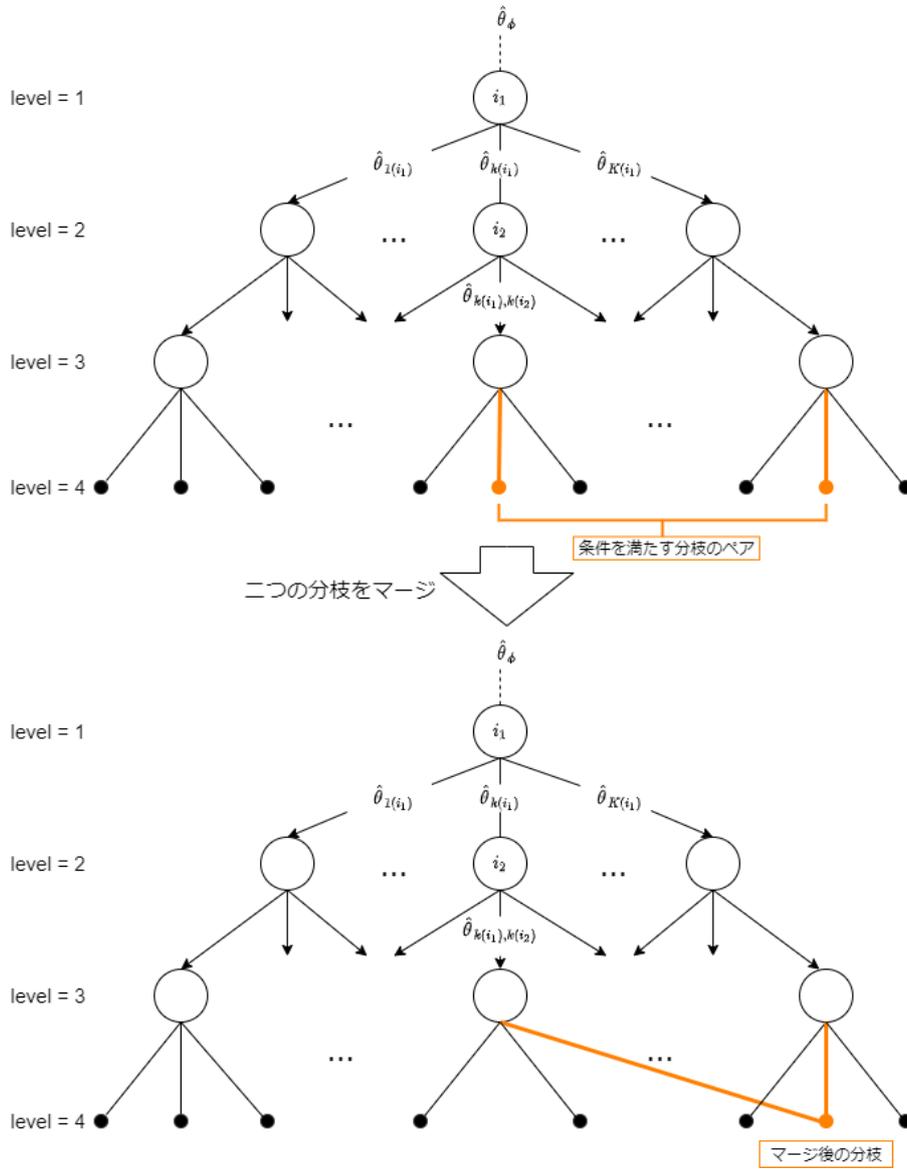


図4 同一階層の分枝のマージ

$$\sum_{n=1}^{Z_{m-1}} K_{i_n} > K^* \quad (5)$$

$$\left| \hat{\theta}_u^{k_s} - \hat{\theta}_v^{k_t} \right| < \frac{L_2 - L_1}{K^*} \quad (6)$$

$$\int_{-\infty}^{\infty} \min \left\{ f_u^{k_s}(\theta), f_v^{k_t}(\theta) \right\} d\theta > \delta \quad (7)$$

条件式 (5) はマージしない状態で階層 m に存在する分枝数の合計があらかじめ設定したパラメータ K^* 以上であるか, 条件式 (6) は二つの分枝に対応する推定能力値が類似しているかどうか, 条件式 (7) は二つの分枝に対応する事後確率分布が類似しているかどうかをそれぞれ判定している.

なお条件 (5) は条件 (7) の判定にかかる計算コストを削減するための条件で, 決定木が十分に成長するまでは条件 (5) と (7) を基にマージを行い, 分枝数があらかじめ定めたパラメータ値を超えた場合は決定木が十分に成長したとみなし, その後は条件 (5) と (6) を基にマージを行う. パラメータ K^* を大きく設定するほど推定精度は上がり, 計算コストは増加する.

Merged Tree-CAT のマージアルゴリズムは *Algorithm 1* に示す. $nodes(m)$ は階層 m で生成される予定の頂点のリストで, それぞれが階層 $m-1$ の頂点からの分岐先に対応している.

マージされる分枝に対応する事後確率分布 $f_{u,v}^{k_s, k_t}$ は次のように, 受験者がそれぞれの分岐先のノードに到達する確率を用いて平均化される.

$$f_{u,v}^{k_s, k_t} = \frac{D_u^{k_s}}{D_u^{k_s} + D_v^{k_t}} + \frac{D_v^{k_t}}{D_u^{k_s} + D_v^{k_t}}$$

また, 受験者がマージ後の分枝に到達する確率 $D_{u,v}^{k_s, k_t}$ と, マージ後の分枝に到達した受験者が回答した項目群 $A_{u,v}^{k_s, k_t}$ は次のように更新される.

$$D_{u,v}^{k_s, k_t} = D_u^{k_s} + D_v^{k_t}$$

$$A_{u,v}^{k_s, k_t} = A_u^{k_s} \cup A_v^{k_t}$$

これらの条件で木の成長を抑制することで, 従来手法に比べ時間・空間計算量の削減に成功した. マージ対象となった二つの頂点の事後確率分布は, 回答者がそれぞれの分枝に到達する確率で平均化される. 推定能力値もマージ後の事後確率分布に従って更新される.

5 提案手法

Merged Tree-CAT は同一階層内の分枝のうち, 推定能力値とその分布が類似するものをマージすることで, 決定木生成に要する時間を削減したが, 大規模アイテムバンクを用いる場合や各項目の回答パターン数が多い場合において, 同一階層に制限しているために

Algorithm 1 Merged Tree-CAT の分枝マージ法

Require: $m, nodes(m), K^*$ **Ensure:** $nodes(m)$

```
1: Initialisation
2:  $length(m) :=$  マージする前の深度  $m$  に存在する分枝数
3: LOOP Process
4: for  $i = 1, \dots, length(m) - 1$  do
5:   for  $j = i + 1, \dots, length(m)$  do
6:     if  $nodes(m)_i$  の推定能力値と  $nodes(m)_j$  の推定能力値が類似 then           ▷ マージ条件 (6)
7:       if  $length(m) > K^*$  then                                           ▷ マージ条件 (5)
8:          $nodes(m)_i$  を  $nodes(m)_j$  にマージ
9:          $nodes(m)_i$  を  $nodes(m)$  から削除
10:        break
11:       else if  $nodes(m)_i$  の推定能力値の分布と  $nodes(m)_j$  の推定能力値の分布が類似 then
12:                                                                                                               ▷ マージ条件 (7)
13:          $nodes(m)_i$  を  $nodes(m)_j$  にマージ
14:          $nodes(m)_i$  を  $nodes(m)$  から削除
15:         break
16:       end if
17:     end if
18:   end for
19: end for
20: return  $nodes(m)$ 
```

以前として決定木生成には時間を要する．そこで本研究では、生成中の階層の上位置の分枝に対してもマージを行うことでさらに決定木を圧縮し、決定木生成時間を軽減する手法を提案する．

具体的には、現在生成中の階層の分枝とその上位置の分枝のペアのうち、上で説明した Merged Tree-CAT のマージ条件 (5)(6) に加え、出題項目の重複を防ぐための条件 (9)、各項目の露出可能率を保つための条件 (8)、決定木内のループを防ぐための条件 (10) を満たす頂点同士を図 5 のようにマージする．

$$C_{i_v}^m \geq D_u^{k_s} \quad (8)$$

$$i_v \notin A_u^{k_s} \quad (9)$$

$$v \notin B_u \quad (10)$$

ここで, i_v を階層 $m-1$ の頂点 v に割り当てられている項目, $A_u^{k_s}$ を階層 $m-1$ の頂点 u で回答 k_s を選択した受験者が今までに回答した項目集合, $c_{i_v}^m$ を階層 $m-1$ 生成後の項目 i_v の露出可能率, $D_u^{k_s}$ を受験者が階層 $m-1$ の頂点 u で回答 k_s を選択する確率, B_u を頂点 u の先祖にあたる頂点集合とする.

また, 受験者がマージ後の頂点 v に到達する確率 D_v と, マージ後の頂点 v に到達して回答 k_t を選択する確率 $D_v^{k_t}$, マージ後の頂点 v に到達した受験者が回答した項目群 A_v , マージ後の頂点 v で回答 k_t を選択した受験者が回答した項目群 $A_v^{k_t}$ は次のように更新される.

$$D_v = D_v + D_u^{k_s}$$

$$D_v^{k_t} = D_v^{k_t} + \frac{D_v^{k_t} D_u^{k_s}}{\sum_{k_t=1}^{K_t} D_v^{k_t}}$$

$$A_v = A_v \cup A_u^{k_s}$$

$$A_v^{k_t} = A_v^{k_t} \cup A_u^{k_s}$$

ここで, D_v を受験者が頂点 v に到達する確率, A_v を頂点 v に到達した受験者が回答した項目群, K_t を項目 t への回答パターン数とする.

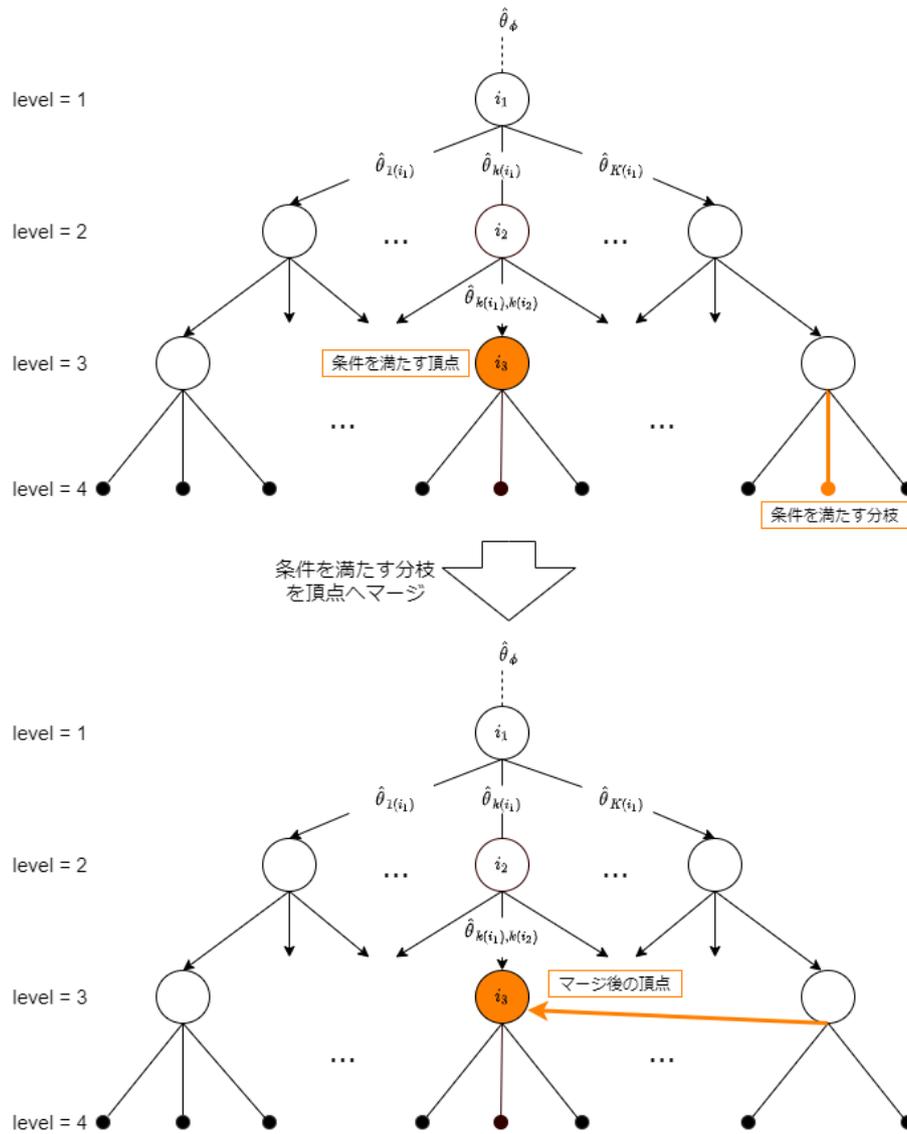


図5 上位の階層の頂点とのマージ

提案手法のアルゴリズムは *Algorithm 2* に示す. Merged Tree-CAT と同様の同一階層内の分枝のマージを行った後, すでに生成した階層 $m-1$ の頂点の中から上で示した条件を満たすものを探索し見つかった場合, 階層 m の分枝をその頂点へマージする.

Algorithm 2 提案手法の分枝マージ法

Require: $m, nodes, K^*$ **Ensure:** $nodes$

```
1: Initialisation
2:  $length(m) :=$  マージする前の階層  $m$  に存在する分枝数
3: LOOP Process
4: for  $i = 1, \dots, length(m) - 1$  do
5:   for  $j = i + 1, \dots, length(m)$  do
6:     if  $nodes(m)_i$ の推定能力値と  $nodes(m)_j$ の推定能力値が類似 then           ▷ マージ条件 (6)
7:       if  $length(m) > K^*$  then                                           ▷ マージ条件 (5)
8:          $nodes(m)_i$ を  $nodes(m)_j$ にマージ
9:          $nodes(m)_i$ を  $nodes(m)$  から削除
10:        break
11:      else if  $nodes(m)_i$ の推定能力値の分布と  $nodes(m)_j$ の推定能力値の分布が類似 then
12:                                                                                          ▷ マージ条件 (7)
13:         $nodes(m)_i$ を  $nodes(m)_j$ にマージ
14:         $nodes(m)_i$ を  $nodes(m)$  から削除
15:        break
16:      end if
17:    end if
18:  end for
19: end for
20: for  $i = 1, \dots, length(m)$  do
21:    $length(m-1) :=$  階層  $m-1$  に存在するノード数
22:   for  $j = 1, \dots, length(m-1)$  do
23:     if  $length(m) > K^*$  then                                           ▷ マージ条件 (5)
24:       if  $nodes(m)_i$ の推定能力値と  $nodes(m-1)_j$ の推定能力値が類似 then
25:                                                                                           ▷ マージ条件 (6)
26:         if  $i_{nodes(m-1)_j} \notin A_{nodes(m)_i}$  then                               ▷ マージ条件 (9)
27:           if  $C_{inodes(m-1)_j}^m \geq D_{nodes(m)_i}$  then                             ▷ マージ条件 (8)
28:             if  $nodes(m-1)_j \notin B_u$  then                                     ▷ マージ条件 (10)
29:                $nodes(m)_i$ を  $nodes(m-1)_j$ へマージ
30:                $nodes(m)_i$ を  $nodes(m)$  から削除
31:             break
32:           end if
33:         end if
34:       end if
35:     end if
36:   end for
37: end for
38: end for
39: return  $nodes$ 
```

6 評価実験

提案手法の有効性を示すため3つの評価実験を行った。実験1では、能力推定精度と決定木生成時間、テスト実施中の項目選択時間に関して従来手法と比較した。比較対象は、決定木を用いた適応型テストである Tree-CAT 法 [3] と Merged Tree-CAT 法 [4]、先行研究で比較されている従来の適応型テスト手法である Restricted 法 [20] の3手法とした。実験2では、決定木の圧縮率と受験者の能力推定精度、決定木生成時間に関して、提案手法と同じく決定木を圧縮する Merged Tree-CAT 法と比較した。実験3では、実験2の結果について要因を分析した。なお、実験環境は Windows10 Pro を OS とする計算機 (CPU: Intel Core i5-8400 2.80GHz, RAM: 48GB) である。

実験には、大きさが 500, 1000, 2000 項目のシミュレーションアイテムバンクと 978 項目の実データからなるアイテムバンクを用いた。シミュレーションアイテムバンクの各項目の識別力パラメータ a は $a \sim \log N(0, 0.1225)$ 、困難度パラメータ b は $b \sim N(0, 1)$ として生成した。実アイテムバンクの詳細は表1の通りである。その他のパラメータは先行研究に従い、 $\theta \sim N(0, 1)$ 、 $p = 0.9$ 、 $\delta = 0.6$ 、 $r_i = 0.3$ とした。

表1 実アイテムバンクの詳細

Pool Size	Parameter a			Parameter b		
	Range	Mean	SD	Range	SD	Mean
978	0.12~3.08	0.43	0.2	-4~4.55	-0.22	1.16

6.1 実験1:従来手法との比較

1000 項目のシミュレーションデータからなるアイテムバンクを用いて、Merged Tree-CAT 法、Tree-CAT 法、Restricted 法との比較実験を行う。Restricted 法は、項目の露出率を制御する通常の CAT 手法の一つで、最大露出率を超えた項目をアイテムバンクから取り除く手法である [20]。各項目の回答パターンは正誤の2つで、識別力パラメータ a を $a \sim \log N(0, 0.1225)$ 、困難度パラメータ b を $b \sim N(0, 1)$ として発生させた [4]。各項目の最大露出率 $r_i = 0.3$ とし、各受験者に出題する項目数を 10 問とした。また、能力推定精度を比較するために、 $\theta \sim N(0, 1)$ から生成した 500 人の受験者の上記の項目への回答パターンを発生させた。実験結果を図6に示す。縦軸を受験者の推定能力値の平均二乗誤差 (MSE: mean squared error)、横軸を出題項目数とする。

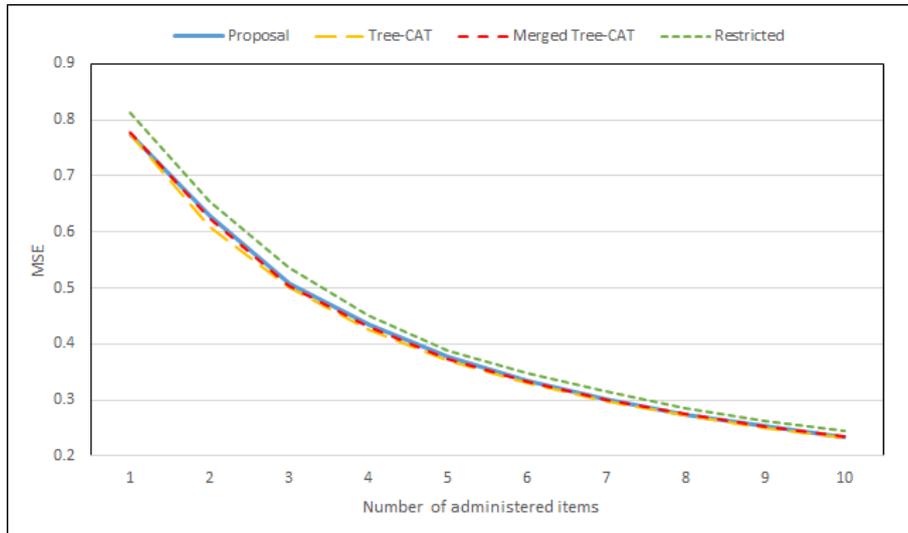


図6 各手法との能力推定精度の比較（シミュレーションデータ）

図6を見ると、Proposal, Tree-CAT, Merged Tree-CATの3手法は同程度の能力推定精度があることが分かる。また、決定木を用いた3つの手法はRestricted法に比べ、推定精度が良いことが分かる。3つの手法の中では推定精度はTree-CAT, Merged Tree-CAT, Proposalの順に高かった。

下の表2は各手法のテスト実施前の決定木生成時間と、テスト実施中の項目選択時間を示している。決定木を用いた適応型テストであるProposal, Tree-CAT, Merged Tree-CATの3手法は、テスト実施前に決定木を生成するため、テスト実施中の項目選択時間が0 secとなっている。それに対しRestricted法は、テスト実施中に出題項目を選択するため、項目選択に平均11.6 secを要した。また、Restricted法は決定木を生成しないため、決定木生成時間はNoneと示した。決定木生成時間を比較すると、Proposal, Merged Tree-CAT, Tree-CATの順に生成時間が短かい、またその中でも、決定木を圧縮するProposalとMerged Tree-CATは、圧縮しないTree-CATに比べ圧倒的に生成時間が短いことがわかる。

表 2 決定木生成時間の比較 (シミュレーションデータ)

Method	決定木の生成時間 [min]	テスト実施中の平均項目選択時間 [sec]
Proposal	2.4	0
Tree-CAT	27.4	0
Merged Tree-CAT	2.8	0
Restricted	None	11.6

続いて 978 項目の実データからなるアイテムバンクを用いて、上と同様の実験を行った。能力推定精度を図 7，決定木生成時間とテスト実施中の項目選択時間を表 3 に示した。シミュレーションデータに比べて、どの手法も推定精度は悪かったが、これは実データの識別力パラメータがシミュレーションデータに比べて小さいことが原因である。その他に関してはシミュレーションデータと同様の結果が得られた。

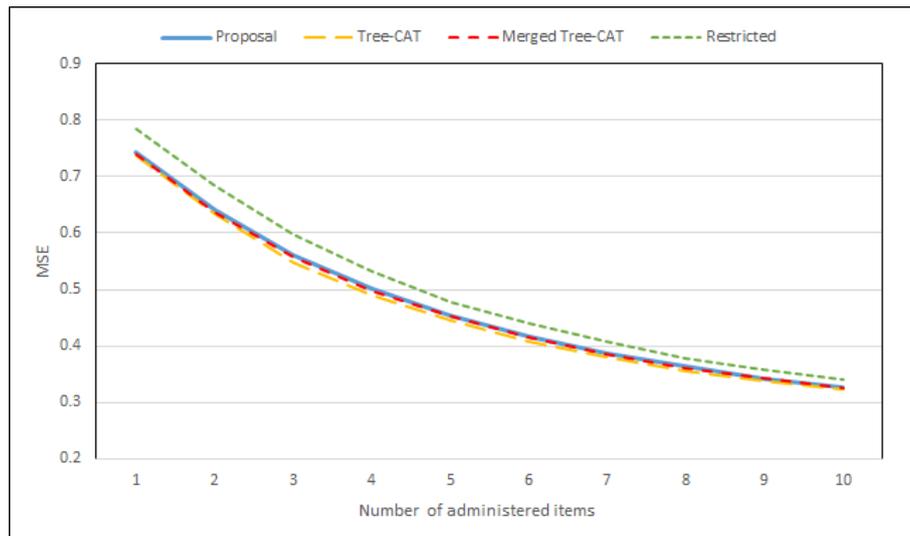


図 7 各手法との能力推定精度の比較 (実データ)

表 3 決定木生成時間の比較 (実データ)

Method	決定木生成時間 [min]	テスト実施中の平均項目選択時間 [sec]
Proposal	2.6	0
Tree-CAT	33.5	0
Merged Tree-CAT	3.1	0
Restricted	None	11.5

6.2 実験 2: Merged Tree-CAT との比較

決定木を圧縮しない Tree-CAT は、分枝数の指数的増加にともない時間・空間計算量が爆発的に増加するため、出題項目数に限界がある。そのため実験 1 の設定の場合、Tree-CAT は 13 項目の出題が限界であった。そこで、さらにアイテムバンクや出題項目数を増やして比較を行うため、実験 2 では提案手法と同じく決定木を圧縮する Merged Tree-CAT のみと比較する。具体的には、大きさが 500,1000,2000 項目のシミュレーションアイテムバンクと 978 項目の実アイテムバンクに対して、 $K^* = 200, 300, 500$ 、出題項目数を 10,30,70 問とした場合の Merged Tree-CAT に対する決定木の圧縮率と推定能力値の平均二乗誤差 (MSE:mean squared error)、決定木生成時間を求めた。ここで、 N_m を Merged Tree-CAT で生成された決定木の頂点数、 N_p を提案手法で生成された決定木の頂点数としたとき、 N_p/N_m を Merged Tree-CAT に対する提案手法の決定木の圧縮率とする。結果はそれぞれ表 4, 5, 6 のようになった。

表 4 決定木の圧縮率

出題項目数			10	30	70
		K^*			
Item Bank	500	200	0.857	0.767	0.797
		300	0.899	0.734	0.788
		500	0.926	0.788	0.789
	1000	200	0.857	0.738	0.782
		300	0.923	0.725	0.725
		500	0.935	0.728	0.711
	2000	200	0.877	0.746	0.786
		300	0.915	0.726	0.721
		500	0.940	0.740	0.703
	978	200	0.839	0.749	0.738
		300	0.922	0.727	0.726
		500	0.926	0.719	0.715

はじめに決定木の圧縮率に関する表 4 をみると、アイテムバンクが大きいほど決定木が圧縮されていることが分かる。これは使用可能な項目が多い方が、提案手法のマージ条件

(8)(9) を満たしやすくなるためである。出題項目数に関しては、全てのシナリオにおいて出題項目数が 10 問の場合が最も圧縮率が低かった。これはテストの序盤において、提案手法のマージ条件 (5) が満たされないことが原因である。また、出題項目数が 10 問の場合は、 K^* の値が小さい方が圧縮率が高く、出題項目数が 70 問の場合は、 K^* の値が大きい方が圧縮率が高い傾向があった。

表 5 能力推定精度の比較

Method			Merged Tree-CAT			Proposal		
出題項目数			10	30	70	10	30	70
		K^*						
Item Bank	500	200	0.247	0.117	0.083	0.249	0.118	0.083
		300	0.245	0.115	0.080	0.246	0.116	0.081
		500	0.243	0.114	0.078	0.244	0.114	0.079
	1000	200	0.234	0.104	0.062	0.235	0.104	0.062
		300	0.233	0.104	0.061	0.233	0.105	0.062
		500	0.231	0.103	0.060	0.231	0.104	0.061
	2000	200	0.226	0.099	0.056	0.227	0.099	0.056
		300	0.224	0.097	0.055	0.225	0.098	0.055
		500	0.223	0.096	0.053	0.223	0.097	0.054
	978	200	0.325	0.178	0.108	0.329	0.180	0.110
		300	0.323	0.175	0.105	0.325	0.176	0.107
		500	0.320	0.173	0.101	0.321	0.175	0.104

受検者の能力推定精度に関する表 5 をみると、全てのシナリオにおいて Merged Tree-CAT が提案手法より高い推定精度をもっているが、その差は非常に小さく、提案手法は Merged Tree-CAT と同程度の推定精度を維持できていることがわかる。

決定木生成時間に関する表 6 を見ると、すべてのシナリオにおいて提案手法の方が短い時間でテストを生成していることが分かる。出題項目数が少ない場合は両手法とも短時間で生成しているため、提案手法の効果はあまり確認できないが、出題項目数が多い場合やアイテムバンクが大きい場合は Merged Tree-CAT を用いても決定木の生成に時間を要していることから、生成時間をさらに削減できる提案手法の有用性が確認できる。また、 K^* の値が大きいほど両手法とも生成に時間を要した。これは、各階層に生成される頂点数が増えるためである。

表 6 テスト生成時間 (min) の比較

Method			Merged Tree-CAT			Proposal		
出題項目数			10	30	70	10	30	70
		K^*						
Item Bank	500	200	1.4	8.4	26.9	1.2	6.5	21.4
		300	2.0	12.4	40.3	1.8	9.1	31.2
		500	2.7	19.5	66.3	2.5	15.4	52.8
	1000	200	2.8	17.2	53.6	2.4	12.7	41.9
		300	4.0	25.1	79.9	3.7	18.2	57.9
		500	5.2	39.3	131.0	5.0	28.6	93.1
	2000	200	5.7	35.8	113.2	5.0	26.7	89.0
		300	7.1	52.0	173.8	6.5	37.6	125.3
		500	9.3	79.4	276.9	9.0	59.2	195.8
	978	200	3.1	18.3	61.4	2.6	13.7	45.3
		300	4.2	26.5	86.3	3.9	19.2	62.6
		500	5.7	43.1	143.5	5.3	31.0	102.6

6.3 実験 3: 圧縮率に関する要因分析

実験 2 の決定木の圧縮率に関する結果で、 K^* の値によって圧縮率に変化が見られた。この要因を調べるため、 $K^* = 200, 500$ の場合に関して、決定木の各階層で生成された頂点数を比較した。実験には項目数 1000 のシミュレーションアイテムバンクを使用した。図 8 は $K^* = 200$ 、図 9 は $K^* = 500$ の結果である。

図 8, 9 をみると、 $K^* = 200$ のときの階層 7 以外は、提案手法の方が生成される頂点が少なく、Merged Tree-CAT に比べ決定木を圧縮できていることが分かる。また、条件 (5) を満たすまでは、提案手法のマージは始まらないため、序盤の階層では頂点数は全く同じになっている。 $K^* = 200$ の階層 7 で例外的に Merged Tree-CAT よりも頂点数が多くなっているのは、一つ前の階層 6 で提案手法のマージを行った結果、次の階層 7 での分枝数が、Merged Tree-CAT のマージ条件 (5) を満たさなくなり、提案手法のマージが行われなかったのが原因である。図 8, 9 を比較すると、 $K^* = 200$ の場合は階層が深くなるにつれ、頂点数の差が急激に縮まっているのに対し、 $K^* = 500$ の場合は階層が深くなっても

一定の差を保っている。

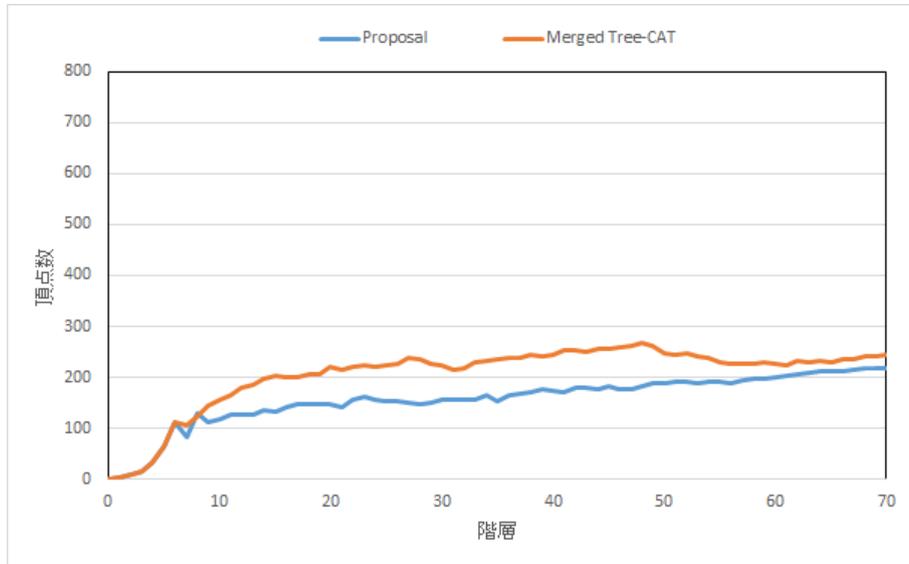


図 8 各階層の頂点数 ($K^* = 200$)

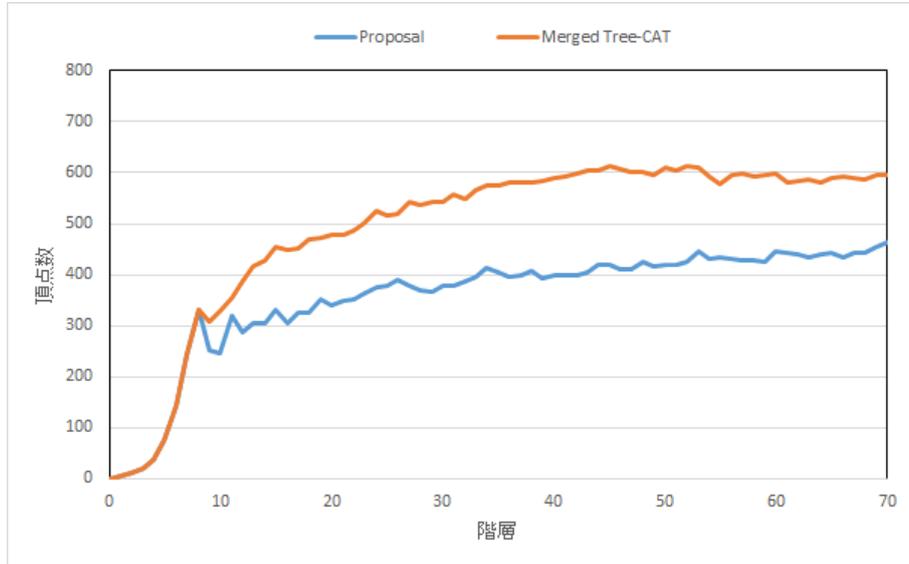


図 9 各階層の頂点数 ($K^* = 500$)

ここで $K^* = 200$ の場合に頂点数の差が急激に縮まる原因を考察する。提案手法のマージ条件 (8)(9)(10) の内、決定木内のループを防ぐための条件 (10) を満たさなくなる数が急激に増えるとは考え難いため、条件 (8)(9) について調べる。

各項目の露出率が設定した最大露出率を超えないようにするための条件 (8) を満たし辛くなっている可能性を調べると、両手法とも階層 70 の決定木生成時点で、アイテムバンク内の 82% の項目が使用され、各項目の平均露出率は 0.07 で、最大露出率 $r_i = 0.3$ には至っておらず、条件 (8) が要因ではないことがわかる。

続いて条件 (9) が満たされていない可能性を調べる。図 10, 11 は $K^* = 200, 500$ の場合について、各階層で推定能力が類似していることを確認する条件 (6) と、一つのテストで出題項目が重複することを防ぐための条件 (9) を満たした回数、実際にマージされた回数を示したグラフである。どちらのグラフも条件 (6) を満たし回数は階層が深くなるにつれ徐々に増えている。序盤の階層で条件 (6)(9) やマージされた回数が 0 になっているのは、序盤の階層では分枝数が条件 (5) を満たさないためである。

また、どちらのグラフも条件 (9) を満たした回数とマージされた回数はほとんど一致している。条件 (9) を満たした回数について、 $K^* = 200, 500$ の場合を比較すると、 $K^* = 200$ の場合は階層 30 辺りから徐々に減り、階層 60 以降は非常に少なくなり、2 回しかマージが行われない階層もあった。それに対し、 $K^* = 500$ の場合は、条件 (9) を満たす回数は徐々に減ってはいるものの、階層が深くなっても一定数の分枝がマージされている。

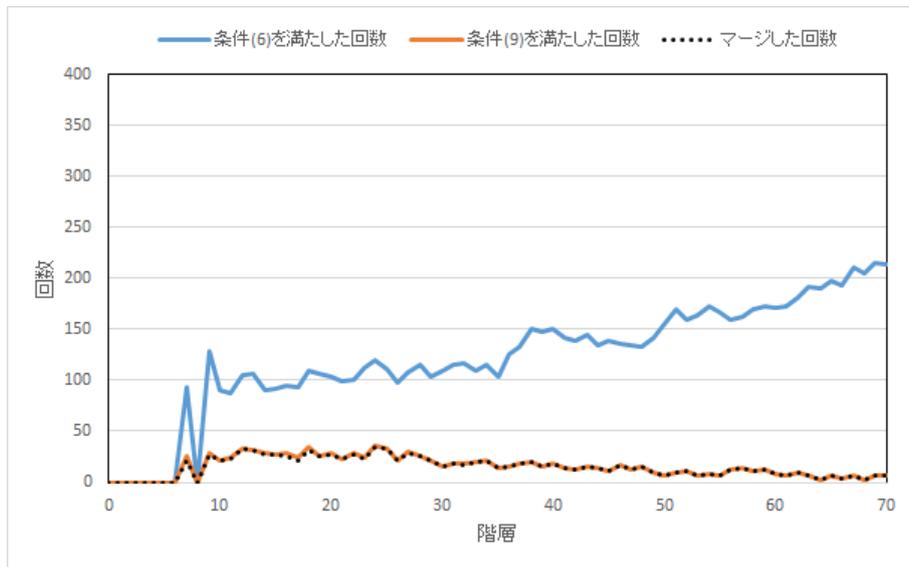


図 10 マージ条件を満たした回数 ($K^* = 200$)

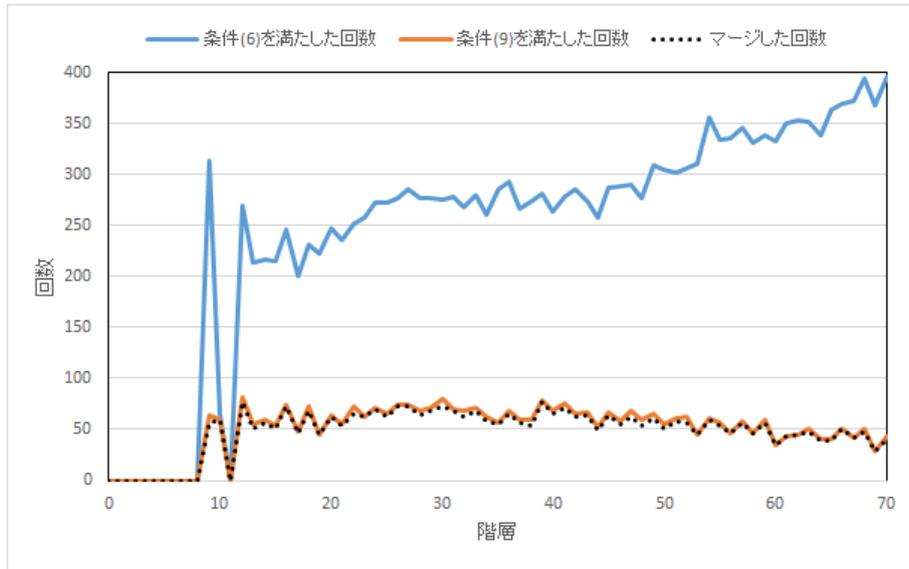


図 11 マージ条件を満たした回数 ($K^* = 500$)

図 10, 11 で, 条件 (9) を満たす回数が減少しているのは, 各分枝に到達した受験者が既に回答した項目群 A が階層が深くなるにつれ膨張ことが要因であると考えられる. 一つのテストで出題項目が重複することを避けるため, 各分枝に到達した受験者が既に回答した項目群が大きくなると, それ以降の階層で出題可能な項目数が減少する. このような出題可能項目数の減少は階層が深くなり, マージが繰り返されることで促進される. 特に出題された項目の重複が少ない分枝同士をマージするほど, マージ後の分枝に対して出題可能な項目数が減少する. このような理由から, 階層が深くなるにつれ提案手法の条件 (9) が満たし難くなり, 結果としてマージ回数が減っているのではないかと考察する. $K^* = 200$ の場合より $K^* = 500$ の場合の方がマージ回数の減少が緩やかなのは, 各階層に存在する頂点数が多いため, 条件 (9) を満たす頂点が見つかり易いためだと思われる. これは条件 (6) を満たした回数の違いからも推察できる.

このように提案手法には, 階層が深くなる, すなわち出題項目数が多くなるにつれ, 各分枝に対して出題可能な項目数が減少し, 結果的にマージ回数が減るという問題があることがわかった. この問題の改善案としては, アイテムバンクを複数の等質アイテムバンクに分割し, 一定の階層数毎に使用するアイテムバンクを切り替えるという手法が考えられる. 上でも述べたように, 提案手法は出題項目の重複が少ない分枝同士ほど, マージ後の分枝に対して出題可能項目が減少してしまう. そこで, 一定の階層数毎にアイテムバンクを分割して使用し, 各階層で出題される項目を意図的に偏らせることで, マージ後の分枝に対して出題可能項目が減少する問題を軽減する.

7 むすび

本論文では決定木を用いた適応型テストのためのテスト生成時間削減手法を提案した。同一階層の分枝の内推定能力値とその分布が類似しているものをマージする Merged Tree-CAT 法に対し、本手法は同一階層内のマージに加え、現在生成中の階層の分枝を、既に生成された上位の階層の頂点へマージすることで、決定木をさらに圧縮した。

実験結果から、提案手法は決定木の圧縮を行わない Tree-CAT 法と比較すると、決定木生成時間を大きく削減できていることがわかった。また、同一階層の分枝のマージのみを行う Merged Tree-CAT 法と比べても劇的な効果は得られなかったものの、受験者の能力推定精度を保ちつつ、決定木生成時間を削減する効果が確認できた。特に Merged Tree-CAT 法でも生成に時間を要する、アイテムバンクが大きい場合や出題項目数が多い場合においては本手法の効果が期待できることがわかった。

また 6.3 で、提案手法には出題項目数が増えると、マージ回数が減ってしまうという課題があり、これは、出題項目の重複が少ない分枝同士をマージするほど、マージ後の分枝に対して出題可能項目が減少するという特性が要因であると考察した。今後は出題項目数によるマージ回数の減少を改善するために、アイテムバンクを複数の等質アイテムバンクに分割し、一定の階層数毎に使用するアイテムバンクを切り替えることで、各階層で出題される項目を意図的に偏らせ、出題項目の重複が少ない分枝同士をマージした場合に生じる、マージ後の分枝に対して出題可能項目が減少する問題を軽減する手法を検討する。

参考文献

- [1] 植野真臣, 永岡慶三, e テスティング, 培風館, 2009.
- [2] M. Ueno, P. Songmuang (2010), “Computerized Adaptive Testing based on Decision Tree” , The 10th IEEE International Conference on Advanced Learning Technologies, pp.191-193.
- [3] D. Delgado-Gomez, Juan C. Laria, Diego Ruiz-Hernandez (2019), “Computerized adaptive test and decision trees: A unifying approach” , Expert Systems With Applications 117 pp.358-366.
- [4] Javier Rodríguez-Cuadrado, David Delgado-Gómez, Juan C. Laria, Sara Rodríguez-Cuadrado (2020), “Merged Tree-CAT: A fast method for building precise computerized adaptive tests based on decision trees” , Expert Systems With Applications 143 pp.113-120
- [5] M. Ueno (2013), “Adaptive Testing Based on Bayesian Decision Theory” , Artificial Intelligence in Education 2013, pp.712-716.
- [6] D. Delgado-Gomez, E. Baca-Garcia, D. Aguado, P. Courtet, J. LopezCastroman (2016), “Computerized Adaptive Test vs. decision trees: Development of a support decision system to identify suicidal behavior” , Journal of Affective Disorders 206 pp.204-209.
- [7] Samejima, F. (2016). Graded response models. In Handbook of Item Response Theory, Volume One, pages 123–136. Chapman and Hall/CRC
- [8] van der Linden, W. J. and Glas, C. A. (2000). Computerized adaptive testing: Theory and practice. Springer
- [9] van der Linden, W. J. and Pashley, P. J. (2009). Item selection and ability estimation in adaptive testing. In Elements of adaptive testing, pages 3–30. Springer. doi:10.1007/978-0-387-85461-8 1.
- [10] van der Linden, W. J. and Veldkamp, B. P. (2005). Constraining item exposure in computerized adaptive testing with shadow tests, volume 2. Law School Admission Council.
- [11] van der Linden, W. J. and Veldkamp, B. P. (2007). Conditional itemexposure control in adaptive testing using item-ineligibility probabili622 ties. Journal of Educational and Behavioral Statistics, 32(4):398–418. doi:10.3102/1076998606298044.
- [12] Sympson, J. and Hetter, R. (1985). Controlling item-exposure rates in computerized adaptive testing. In Proceedings of the 27th annual meeting of the Military Testing As-

sociation, pages 973–977.

- [13] Chang, H.-H. and Ying, Z. (1996). A global information approach to computerized adaptive testing. *Applied Psychological Measurement*, 20(3):213–229. doi:10.1177/014662169602000303
- [14] Lord, F. M. (1980). *Applications of item response theory to practical testing problems*. Hillsdale, NJ: Lawrence Erlbaum.
- [15] Weiss, D. J. (1982). Improving measurement quality and efficiency with adaptive testing. *Applied psychological measurement*, 6(4):473–492. doi:10.1177/014662168200600408.
- [16] Veerkamp, W. J. and Berger, M. P. (1997). Some new item selection criteria for adaptive testing. *Journal of Educational and Behavioral Statistics*, 22(2):203–226. doi:10.3102/10769986022002203
- [17] Weissman, A. (2007). Mutual information item selection in adaptive classification testing. *Educational and Psychological Measurement*, 67(1):41–58. doi:10.1177/0013164406288164.
- [18] Kingsbury, G. G. and Zara, A. R. (1989). Procedures for selecting items for computerized adaptive tests. *Applied measurement in education*, 2(4):359–375. doi:10.1207/s15324818ame02046.
- [19] van der Linden, W. J. (2003). Some alternatives to sympon-hetter item exposure control in computerized adaptive testing. *Journal of Educational and Behavioral Statistics*, 28(3):249–265. doi:10.3102/10769986028003249.
- [20] Revuelta, J. and Ponsoda, V. (1998). A comparison of item exposure control methods in computerized adaptive testing. *Journal of Educational Measurement*, 35(4):311–327. doi:10.1111/j.1745-3984.1998.tb00541.x