

複数テストフォームの構成

加藤健太郎

ベネッセ教育総合研究所 アセスメント研究開発室

日本テスト学会ワークショップ

『項目反応理論を用いたテスト運用の実際』

2014/12/21

本発表の話題

- テスト構成について
 - 自動テスト構成の概要
 - 複数フォームの構成法
- ごくごく簡単な計算例
 - 線形計画法を用いた複数フォームの構成
 - 逐次構成, 同時構成, BSTの比較

テスト構成とは？

テスト構成 (test assembly)

- 項目プールから、テスト仕様に適合するように、テストに出題する項目を選定する作業
 - つまり、「フォーム(テスト冊子)」を作ること
 - 項目プールからの「アウトプット」の部分
- テスト仕様・・・各フォームが満たしているべき条件のリスト
 - (領域ごと／全体の)項目数
 - 解答時間
 - 難易度
 - 測定精度・判定精度
 - ...
- 前提・・・適切な項目プール(項目データベース)があること
 - 選定候補となる項目がすべて共通尺度に乗っている(等化済みである)
 - 選定候補となる項目に、テスト仕様に対応した属性がついている(項目パラメタ, 領域, 解答時間, etc.)
 - 候補項目は、事前の利用計画にもとづいて選定されている

自動テスト構成 (automatic test assembly: ATA)

- 数値的最適化によるテスト構成
 - テスト構成は様々な条件や計算を含む複雑な作業→自動化
 - 客観的な基準に照らして、構成されたテストの出来栄を評価できる
 - 各手法については後述

複数フォームの構成

- IRTを利用したテストにおいて、単一のフォームを使い続けることはほぼあり得ない(IRTを使う意味がない！)
 - 等質なフォーム
 - 異なる受験対象や目的に対応したフォーム
- 構成作業はより複雑に
 - 項目プール中の項目全体の効率的な配置
 - 「手作業」はさらに困難→ATAが絶大な力を発揮

複数テストの構成（線形計画法）

- van der Linden (2005, chap. 6)
- 逐次構成 (sequential construction)
 - その都度ベストな組み合わせ→後になるほどエラー（目標値とのズレ）が増大
- 同時構成 (simultaneous construction)
 - 項目プールのサイズand/or構成するフォーム数が大きくなると、計算コストが増大
- Big Shadow Test (BST)
 - 計算コストと性能のバランスを考えるとgoodな選択
 - フォーム間での項目の重複を扱うのが困難

複数テストの構成(より高度な方法)

- 大量の等質テストの同時構成
 - エラーの均一化・最小化 \longleftrightarrow 計算量とのトレードオフ
 - 項目プールから抽出するフォーム数の最大化(理論的な裏付け)
 - フォーム間での項目の重複
- Belov & Armstrong (2006)
 - 最大集合充填アルゴリズム
- Sun, et al. (2008)
 - 遺伝的アルゴリズム
- Songmuang & Ueno (2011)
 - Beesアルゴリズム(ランダムサーチ法 & 並列計算)
- Ishii, Songmuang, & Ueno (2014)
 - 最大クリーク抽出アルゴリズム(+ランダムサーチ)

計算例

項目プール

- 2パラメタ・ロジスティックモデル
- 現実の項目

項目数(J)	識別力(a)			困難度(b)		
	1Q	Med	3Q	1Q	Med	3Q
50	0.63	0.86	1.10	-0.96	-0.34	0.70

テスト仕様

- 個々のフォームについて
 - 項目数=10
 - 目標テスト情報量(TIF)を指定
 - 他の制約はつけない
- フォーム間で
 - TIFに関して等質な4つのフォームを作る
 - 項目の重複は認めない

線形計画法

(linear programming: LP)

- 線形計画問題 (LP problem)
「 $\mathbf{Ax}=\mathbf{b}$, $\mathbf{l}\leq\mathbf{b}\leq\mathbf{u}$, $\mathbf{l}_x\leq\mathbf{x}\leq\mathbf{u}_x$ という条件の下で, $z=\mathbf{c}^T\mathbf{x}$ を最大化(最小化)する」
 - \mathbf{x} : 決定変数
 - \mathbf{A} , \mathbf{b} , \mathbf{l} , \mathbf{u} , \mathbf{l}_x , \mathbf{u}_x : 制約条件 (所与)
 - z : 目的関数 (係数 \mathbf{c} は所与)
- 整数計画法 (integer programming: IP) 「組み合わせ最適化」の問題
 - \mathbf{x} の要素がすべて整数
- 0-1整数計画法 (0-1 integer programming)
 - \mathbf{x} の要素がすべて0または1の二値
- 混合整数計画法 (mixed integer programming: MIP)
 - \mathbf{x} の要素に整数と実数が混在
- 参考書: 茨木・福島(1993, 1~3章)など

線形計画法

- ATAでは、主に0-1整数計画問題または混合整数計画問題を扱う
- 0-1整数計画問題
 - $\mathbf{x}=[x_1 \ x_2 \ \dots \ x_J]^T$, $x_j=0, 1$
 - 項目*j*がフォームに含まれる場合は $x_j=1$, そうでなければ $x_j=0$
 - J は項目プールのサイズ
 - 参考書:加藤・山田・川端(2014, 15章)
- 混合整数計画問題
 - $\mathbf{x}=[x_1 \ x_2 \ \dots \ x_J \ \mathbf{y}]^T$, $x_j=0, 1$, $\mathbf{y} \geq 0$
 - 項目*j*がフォームに含まれる場合は $x_j=1$, そうでなければ $x_j=0$
 - y はエラーの量(後述)

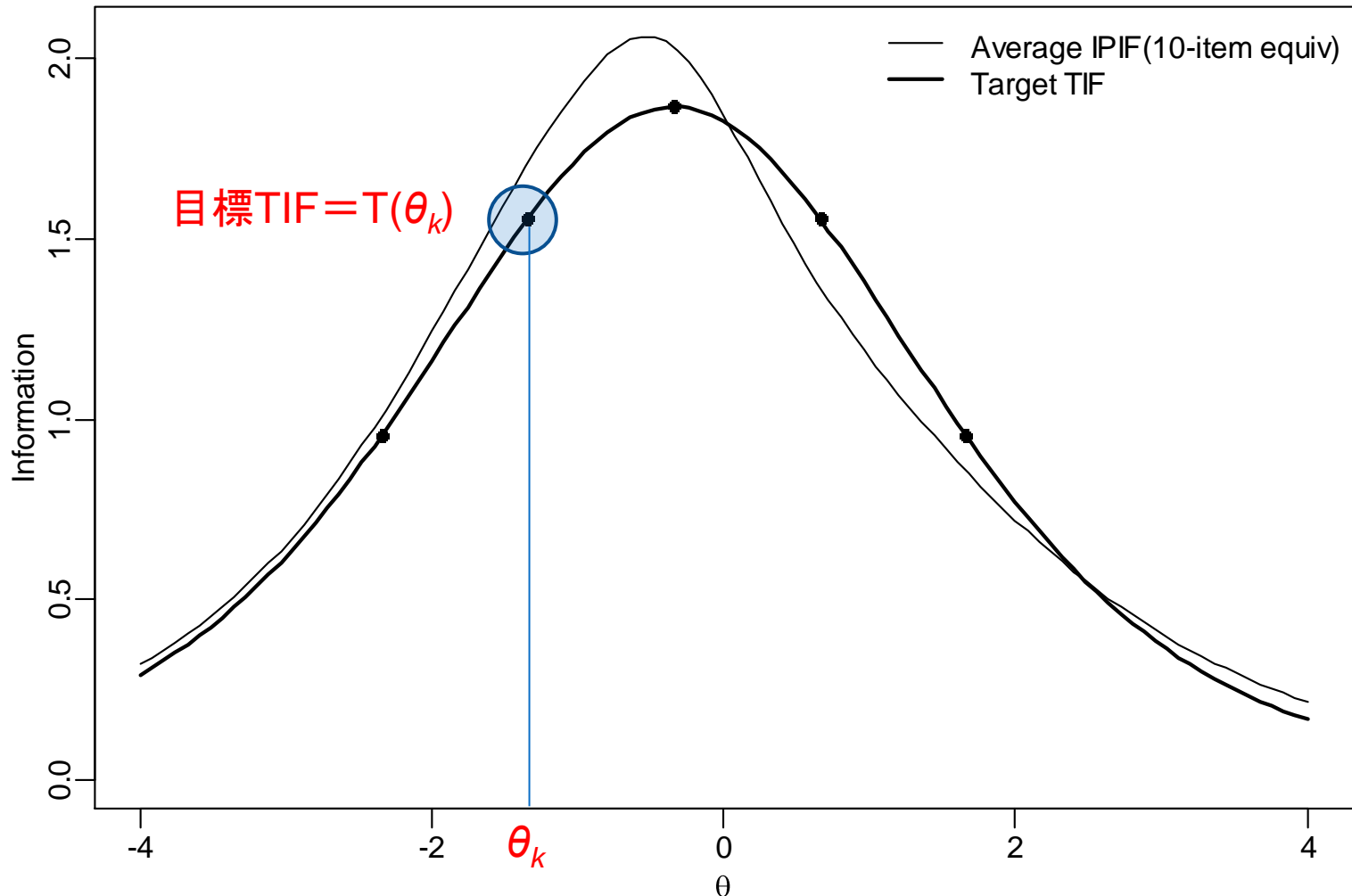
テスト仕様のモデル化

- テスト仕様を，線形計画問題として扱えるように，制約条件および目的関数の形に焼き直す
- 制約条件：目標TIF

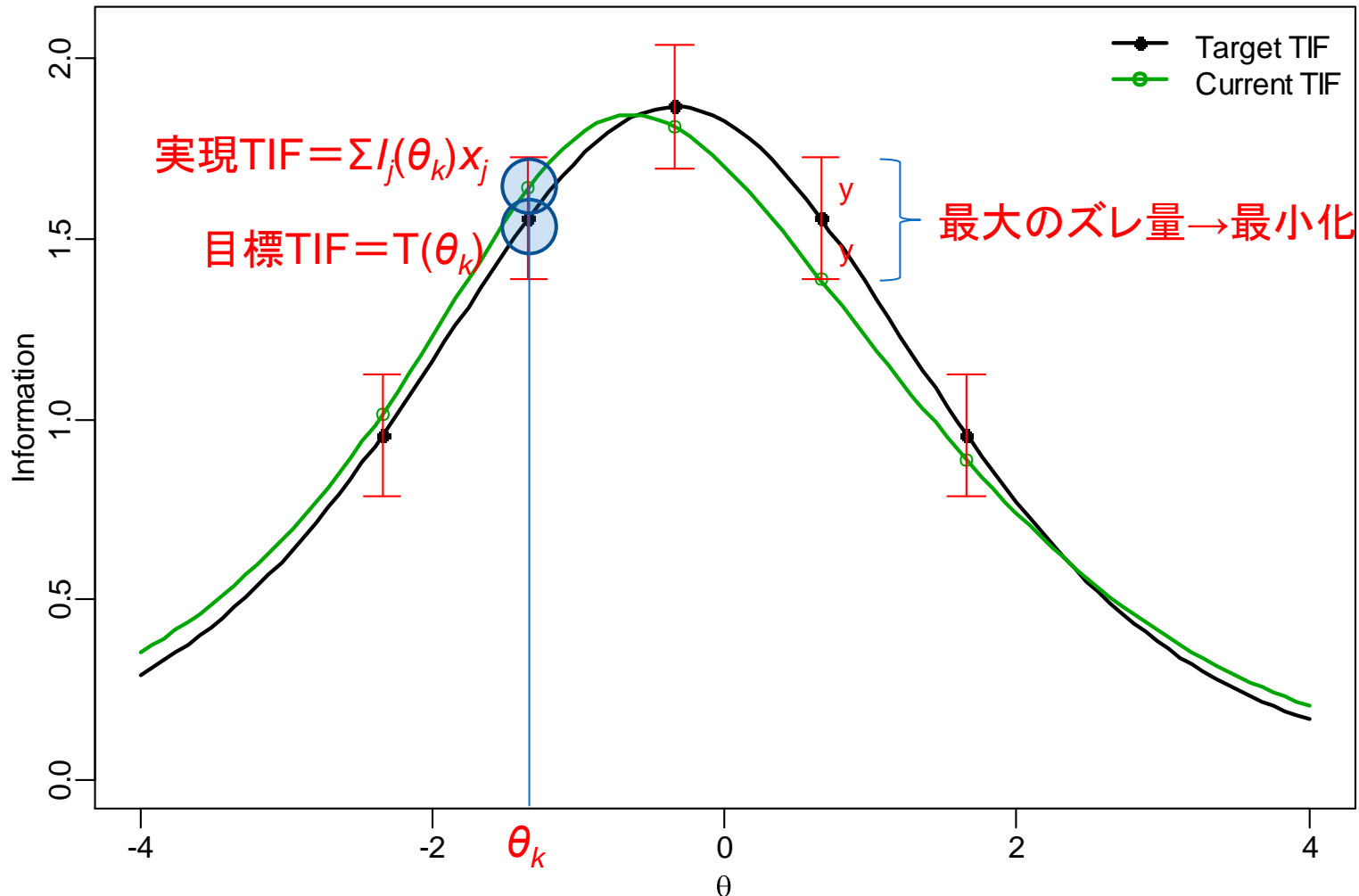
$$\left| \sum_{j=1}^J I_j(\theta_k) x_j - T(\theta_k) \right| \leq y, \quad \forall k = 1, \dots, K$$
$$\Leftrightarrow \begin{cases} \sum_{j=1}^J I_j(\theta_k) x_j + y \geq T(\theta_k) \\ \sum_{j=1}^J I_j(\theta_k) x_j - y \leq T(\theta_k) \end{cases} \quad \forall k = 1, \dots, K$$

- $y \geq 0$ は各評価点 θ_k での目標TIFと実現TIFとのズレ(エラー)の最大量
- 目的関数： $z = y \rightarrow$ 最小化 (minimaxアプローチ)
- R + glpkAPIパッケージで計算

項目プールの平均情報量と 目標テスト情報量



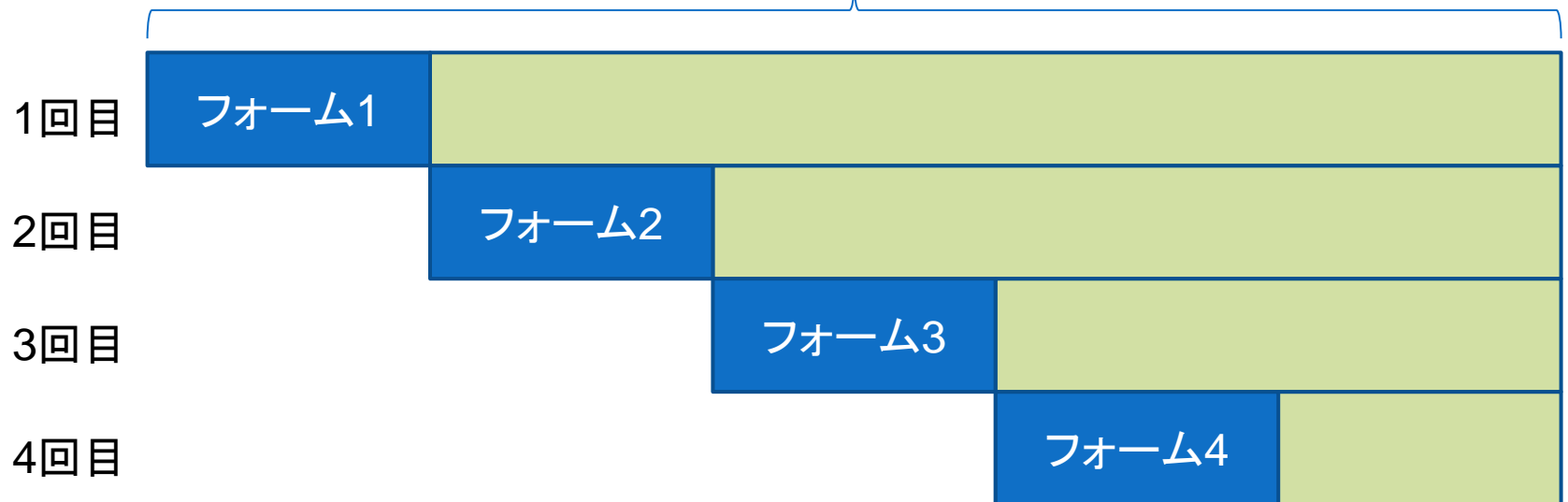
目標TIFとminimaxアプローチ



逐次構成: 考え方

- 1フォームずつ構成
 - 構成するフォームの数だけ計算を実行
 - 都度, 既にフォームに採用された項目は候補から除外

項目プール



逐次構成: テスト構成モデル

$$\mathbf{x}_{(J+1) \times 1} = [x_1 \ \cdots \ x_J \ y]^T, \quad x_j \in \{0,1\}, \quad y \geq 0$$

$$z = \mathbf{c}^T \mathbf{x} = [0 \ \cdots \ 0 \ 1]^T \mathbf{x} = y \rightarrow \text{最小化}$$

項目 $j(x_j)$ に
対応する列

y に
対応する列

項目数

テスト
情報量
(下限)

$$\mathbf{l}_{(2K+1) \times 1} =$$

$$\mathbf{A}_{(2K+1) \times (J+1)} =$$

$$\mathbf{u}_{(2K+1) \times 1} =$$

テスト
情報量
(上限)

10	1	...	1	...	1	0	10
$T(\theta_1)$	$I_1(\theta_1)$...	$I_j(\theta_1)$...	$I_J(\theta_1)$	1	∞
\vdots	\vdots	\ddots			\vdots	\vdots	\vdots
$T(\theta_k)$	$I_1(\theta_k)$...	$I_j(\theta_k)$...	$I_J(\theta_k)$	\vdots	\vdots
\vdots	\vdots	\ddots			\vdots	\vdots	\vdots
$T(\theta_K)$	$I_1(\theta_K)$...	$I_j(\theta_K)$...	$I_J(\theta_K)$	1	∞
$-\infty$	$I_1(\theta_1)$...	$I_j(\theta_1)$...	$I_J(\theta_1)$	-1	$T(\theta_1)$
\vdots	\vdots	\ddots			\vdots	\vdots	\vdots
\vdots	$I_1(\theta_k)$...	$I_j(\theta_k)$...	$I_J(\theta_k)$	\vdots	$T(\theta_k)$
\vdots	\vdots	\ddots			\vdots	\vdots	\vdots
$-\infty$	$I_j(\theta_K)$...	$I_j(\theta_K)$...	$I_J(\theta_K)$	-1	$T(\theta_K)$

J項目分

逐次構成: **glpkAPI**における指定

決定変数のサイズとラベル

```
addColsGLPK(prob, J+1) # 列数=項目数+1
```

```
setColsNamesGLPK(prob, 1:(J+1), c(paste0("x", 1:J), "y")) # 変数名
```

決定変数の型指定

```
setColsKindGLPK(prob, 1:J, rep(GLP_BV, J)) # xjは二値型
```

```
setColKindGLPK(prob, J+1, GLP_CV) # yは連続型
```

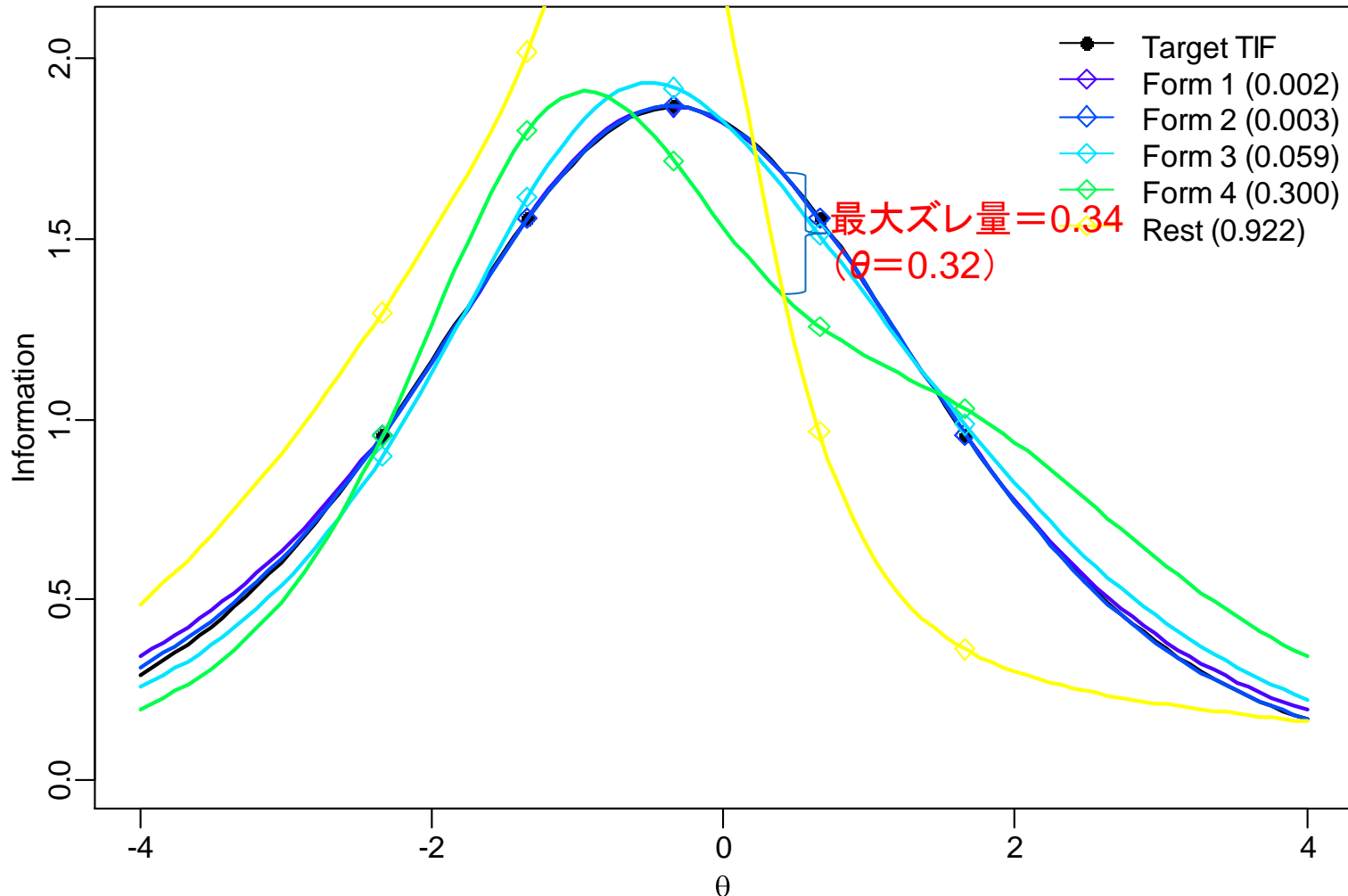
決定変数の制約

```
setColBndGLPK(prob, J+1, GLP_LO, 0.0, 0.0) # y ≥ 0
```

係数cの指定

```
setObjCoefsGLPK(prob, 1:(J+1), c(rep(0, J), 1))
```

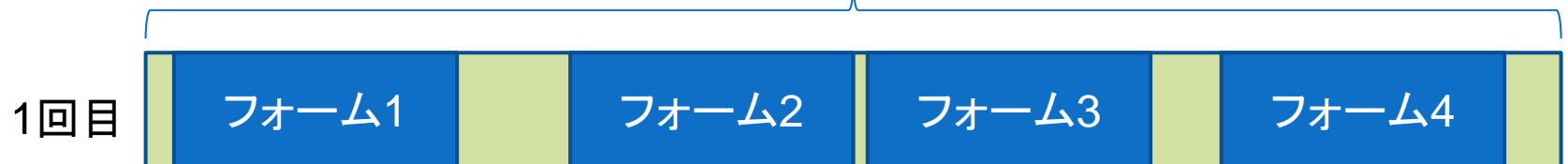
逐次構成: 結果



同時構成: 考え方

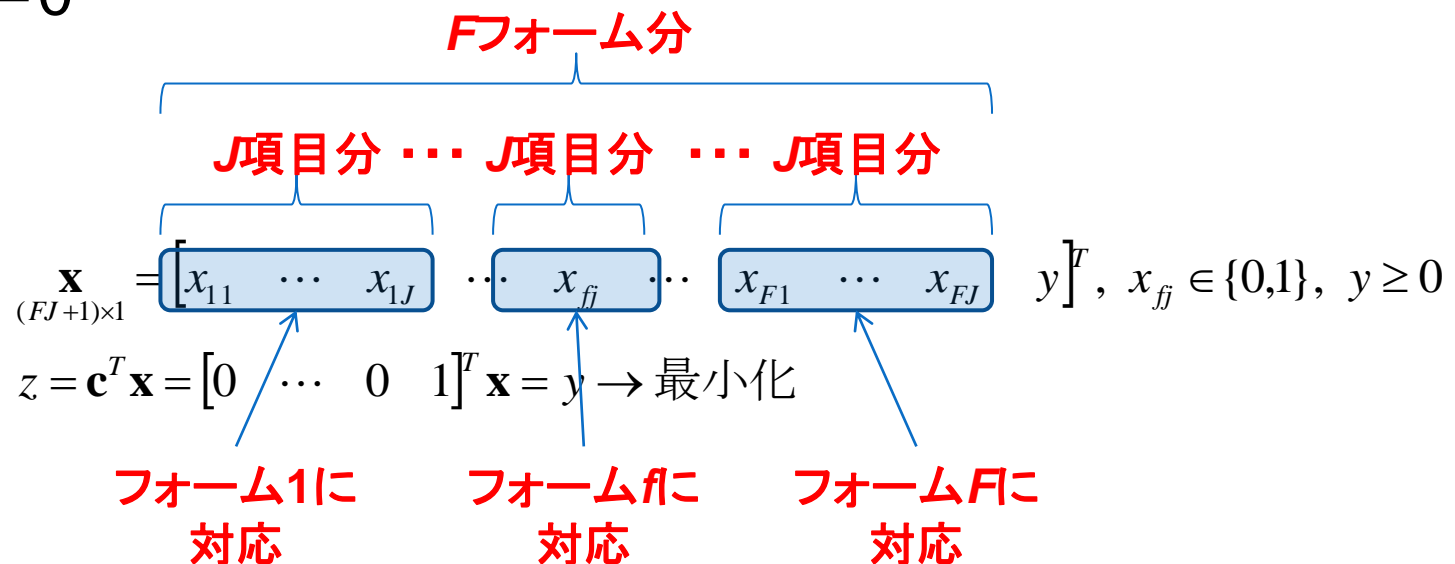
- すべてのフォームを一度に構成
 - 構成するフォームについて, 全体最適化を目指す
 - 計算の実行は1回だが, 規模は増大

項目プール



同時構成: テスト構成モデル

- 決定変数ベクトル \mathbf{x} をフォームの数だけ「拡張」
 - 項目 j がフォーム f に含まれる場合は $x_{fj}=1$, そうでなければ $x_{fj}=0$



- 制約条件も, \mathbf{x} に対応して拡張する

同時構成: テスト構成モデル

- 項目数

フォームfの
項目数制約

$$\mathbf{1}_J^T = [1 \ 1 \ \dots \ 1], \quad \mathbf{0}_J^T = [0 \ 0 \ \dots \ 0]$$

$$\mathbf{1}_0 = \begin{bmatrix} 10 \\ 10 \\ \vdots \\ 10 \end{bmatrix}_{F \times 1}, \quad \mathbf{A}_0 = \begin{bmatrix} \mathbf{1}_J^T & \mathbf{0}_J^T & \dots & \mathbf{0}_J^T & 0 \\ \mathbf{0}_J^T & \mathbf{1}_J^T & \dots & \mathbf{0}_J^T & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_J^T & \mathbf{0}_J^T & \dots & \mathbf{1}_J^T & 0 \end{bmatrix}_{F \times (FJ+1)}, \quad \mathbf{u}_0 = \begin{bmatrix} 10 \\ 10 \\ \vdots \\ 10 \end{bmatrix}_{F \times 1}$$

フォームfに
対応する列
(J項目分)

yに
対応する列

同時構成: テスト構成モデル

- 目標TIF

$$\mathbf{1}^*_{2K \times 1} = \begin{bmatrix} T(\theta_1) \\ \vdots \\ T(\theta_k) \\ \vdots \\ T(\theta_K) \\ -\infty \\ \vdots \\ \vdots \\ \vdots \\ -\infty \end{bmatrix}, \quad \mathbf{A}^*_{2K \times J} = \begin{bmatrix} I_1(\theta_1) & \cdots & I_j(\theta_1) & \cdots & I_J(\theta_1) \\ \vdots & \ddots & & & \vdots \\ I_1(\theta_k) & & I_j(\theta_k) & & I_J(\theta_k) \\ \vdots & & & \ddots & \vdots \\ I_1(\theta_K) & \cdots & I_j(\theta_K) & \cdots & I_J(\theta_K) \\ I_1(\theta_1) & \cdots & I_j(\theta_1) & \cdots & I_J(\theta_1) \\ \vdots & \ddots & & & \vdots \\ I_1(\theta_k) & & I_j(\theta_k) & & I_J(\theta_k) \\ \vdots & & & \ddots & \vdots \\ I_1(\theta_K) & \cdots & I_j(\theta_K) & \cdots & I_J(\theta_K) \end{bmatrix}, \quad \mathbf{u}^*_{2K \times 1} = \begin{bmatrix} \infty \\ \vdots \\ \vdots \\ \vdots \\ \infty \\ T(\theta_1) \\ \vdots \\ T(\theta_k) \\ \vdots \\ T(\theta_K) \end{bmatrix}, \quad \mathbf{a}^*_{2K \times 1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ \vdots \\ 1 \\ -1 \\ -1 \\ \vdots \\ \vdots \\ -1 \end{bmatrix}$$

同時構成: テスト構成モデル

- 目標TIF

フォーム*f*の
情報量
(上限・下限)

$$\mathbf{I}_1 = \begin{bmatrix} \mathbf{I}^* \\ \mathbf{I}^* \\ \vdots \\ \mathbf{I}^* \end{bmatrix}_{2FK \times 1}, \quad \mathbf{A}_1 = \begin{bmatrix} \mathbf{A}^* & \mathbf{O} & \dots & \mathbf{O} & \mathbf{a}^* \\ \mathbf{O} & \mathbf{A}^* & \dots & \mathbf{O} & \mathbf{a}^* \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{O} & \mathbf{O} & \dots & \mathbf{A}^* & \mathbf{a}^* \end{bmatrix}, \quad \mathbf{u}_1 = \begin{bmatrix} \mathbf{u}^* \\ \mathbf{u}^* \\ \vdots \\ \mathbf{u}^* \end{bmatrix}_{2FK \times 1}$$

フォーム*f*に
対応する列
(*J*項目分)

*y*に
対応する列

フォーム内の制約は、対角
ブロック行列として表現

同時構成: テスト構成モデル

- フォーム間で項目の重複を許さないための制約を追加

$$\mathbf{I}_J = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

フォーム間の制約は,
「行」ブロック行列として表現

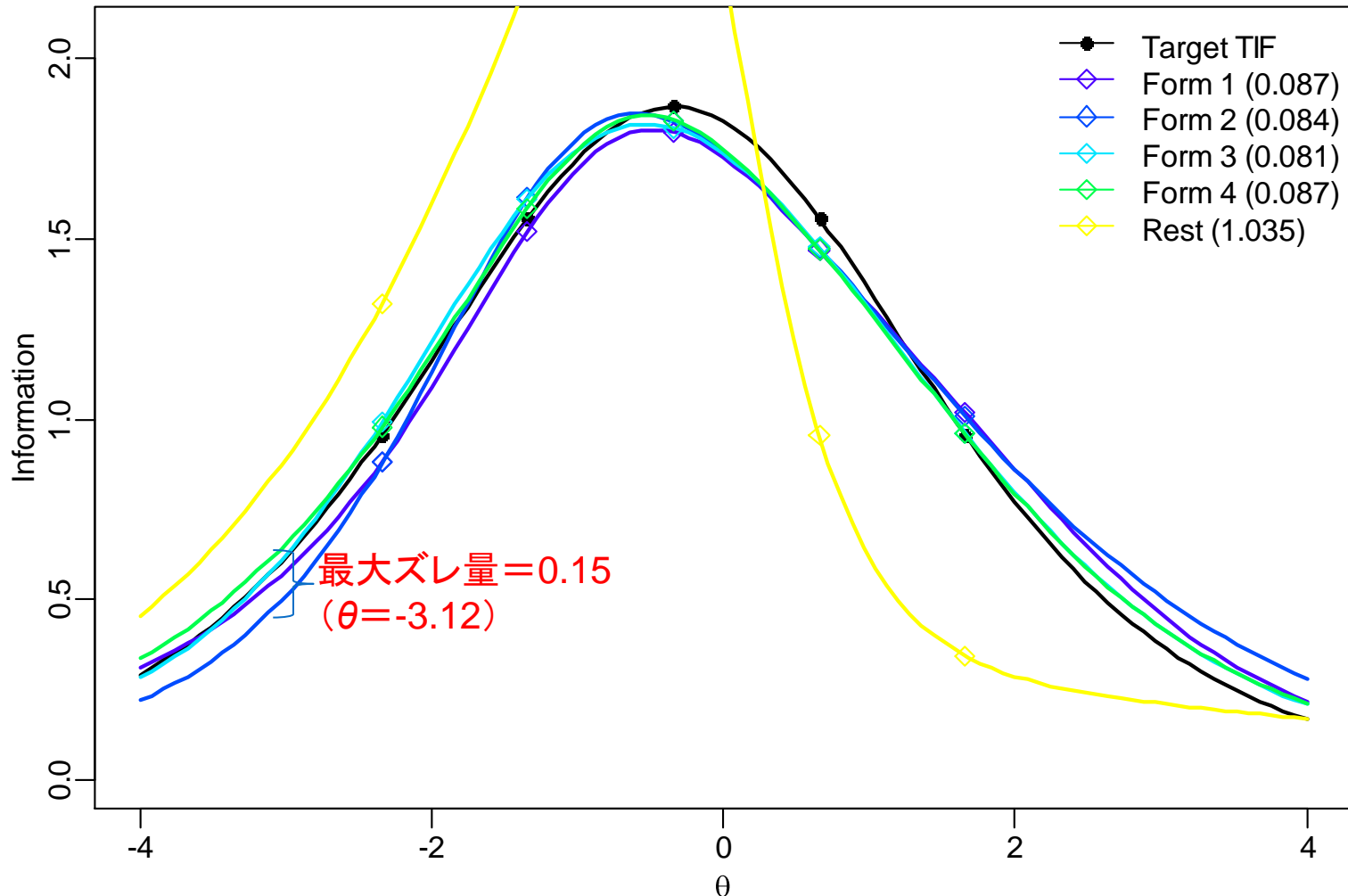
$$\mathbf{l}_2 = \mathbf{0}_J, \quad \mathbf{A}_2 = \begin{bmatrix} \mathbf{I}_J & \mathbf{I}_J & \cdots & \mathbf{I}_J & \mathbf{0}_J \end{bmatrix}, \quad \mathbf{u}_2 = \mathbf{1}_J$$

フォーム f に
対応する列
(J 項目分)
 y に
対応する列

- まとめると...

$$\mathbf{l} = \begin{bmatrix} \mathbf{l}_0 \\ \mathbf{l}_1 \\ \mathbf{l}_2 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}$$

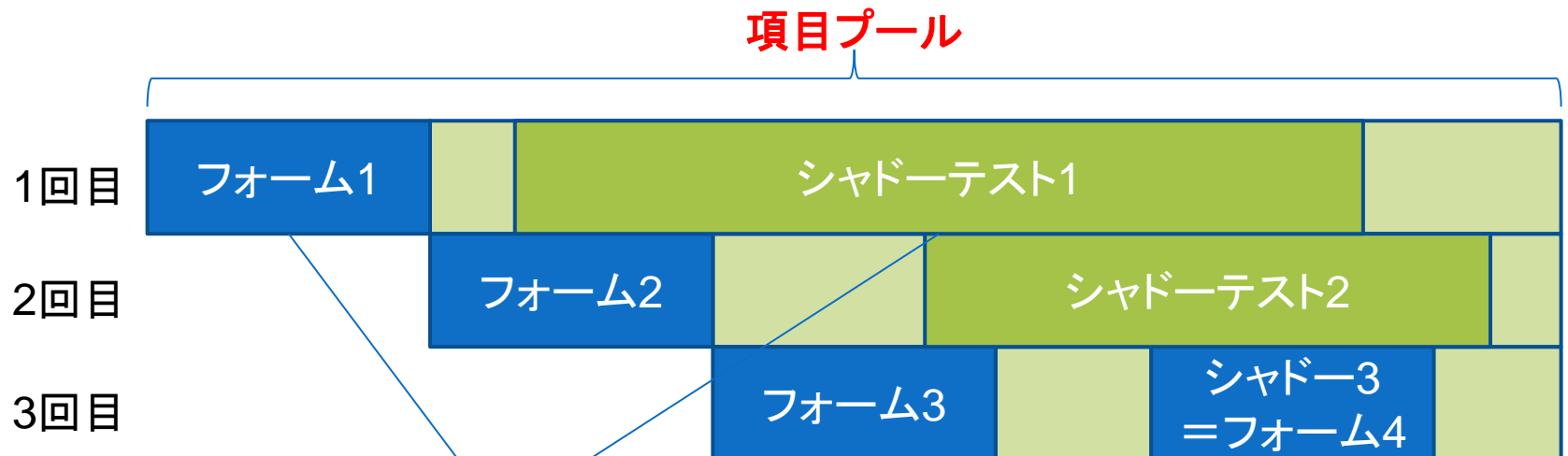
同時構成: 結果



BST: 考え方

- 逐次構成と同時構成の折衷案

- 毎回、「構成するフォーム」「将来構成するフォームの候補(= shadow test)」の2つのフォームを同時構成→これを繰り返す



フォームとシャドーのそれぞれに目標TIFを設定し、都度2フォームを同時構成

BST: テスト構成モデル

- 決定変数ベクトル \mathbf{x} を構成フォーム+シャドーテストに拡張
 - 項目 j がフォーム $f(=F$ または $S)$ に含まれる場合は $x_{fj}=1$, そうでなければ $x_{fj}=0$

2フォーム分

J項目分 J項目分

$$\mathbf{x}_{(2J+1) \times 1} = [x_{F1} \ \cdots \ x_{FJ} \quad x_{S1} \ \cdots \ x_{SJ} \quad y]^T, \quad x_{fj} \in \{0,1\}, \quad y \geq 0$$
$$z = \mathbf{c}^T \mathbf{x} = [0 \ \cdots \ 0 \quad 1]^T \mathbf{x} = y \rightarrow \text{最小化}$$

構成するフォーム
に対応

シャドーテスト
に対応

- 制約条件も, \mathbf{x} に対応して拡張する

BST: テスト構成モデル

- 項目数

構成フォームの
項目数制約

$$\mathbf{1}_J^T = [1 \ 1 \ \dots \ 1], \quad \mathbf{0}_J^T = [0 \ 0 \ \dots \ 0]$$

$\mathbf{1}_0 =$	$[10]$	$\mathbf{A}_0 =$	$\mathbf{1}_I^T$	$\mathbf{0}_I^T$	0	$\mathbf{u}_0 =$	$[10]$
2×1	$10F_{rest}$	$2 \times (FJ + 1)$	$\mathbf{0}_J^T$	$\mathbf{1}_J^T$	0	2×1	$10F_{rest}$

シャドーテスト
の項目数制約

F_{rest} = 残りの構成フォーム数

フォーム f に y に
対応する列 対応する列
(J 項目分)

BST: テスト構成モデル

- 目標TIF

$$\mathbf{1}^*_{2K \times 1} = \begin{bmatrix} T(\theta_1) \\ \vdots \\ T(\theta_k) \\ \vdots \\ T(\theta_K) \\ -\infty \\ \vdots \\ \vdots \\ \vdots \\ -\infty \end{bmatrix}, \quad \mathbf{A}^*_{2K \times J} = \begin{bmatrix} I_1(\theta_1) & \cdots & I_j(\theta_1) & \cdots & I_J(\theta_1) \\ \vdots & \ddots & & & \vdots \\ I_1(\theta_k) & & I_j(\theta_k) & & I_J(\theta_k) \\ \vdots & & & \ddots & \vdots \\ I_1(\theta_K) & \cdots & I_j(\theta_K) & \cdots & I_J(\theta_K) \\ I_1(\theta_1) & \cdots & I_j(\theta_1) & \cdots & I_J(\theta_1) \\ \vdots & \ddots & & & \vdots \\ I_1(\theta_k) & & I_j(\theta_k) & & I_J(\theta_k) \\ \vdots & & & \ddots & \vdots \\ I_1(\theta_K) & \cdots & I_j(\theta_K) & \cdots & I_J(\theta_K) \end{bmatrix}, \quad \mathbf{u}^*_{2K \times 1} = \begin{bmatrix} \infty \\ \vdots \\ \vdots \\ \vdots \\ \infty \\ T(\theta_1) \\ \vdots \\ T(\theta_k) \\ \vdots \\ T(\theta_K) \end{bmatrix}, \quad \mathbf{a}^*_{2K \times 1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ \vdots \\ 1 \\ -1 \\ -1 \\ \vdots \\ \vdots \\ -1 \end{bmatrix}$$

BST: テスト構成モデル

- 目標TIF

構成フォーム
の情報量
(上限・下限)
シャドーテスト
の情報量
(上限・下限)

$$\begin{array}{c}
 \mathbf{l}_1 = \left[\begin{array}{c|c|c|c|c} \mathbf{l}^* & & \mathbf{A}^* & \mathbf{O} & \mathbf{a}^* \\ \hline \mathbf{F}_{rest} & \mathbf{l}^* & \mathbf{O} & \mathbf{A}^* & w_{rest} \mathbf{a}^* \\ \hline \mathbf{u}_1 = & & & & \mathbf{u}^* \\ \hline \mathbf{F}_{rest} & & & & \mathbf{u}^* \end{array} \right] \\
 \mathbf{A}_1 = \left[\begin{array}{c|c|c|c|c} & & \mathbf{A}^* & \mathbf{O} & \mathbf{a}^* \\ \hline & & \mathbf{O} & \mathbf{A}^* & w_{rest} \mathbf{a}^* \\ \hline & & & & \mathbf{u}^* \\ \hline & & & & \mathbf{u}^* \end{array} \right] \\
 \mathbf{u}_1 = \left[\begin{array}{c|c|c|c|c} & & & & \mathbf{u}^* \\ \hline & & & & \mathbf{u}^* \\ \hline & & & & \mathbf{u}^* \\ \hline & & & & \mathbf{u}^* \end{array} \right]
 \end{array}$$

F_{rest} = 残りの構成フォーム数, w_{rest} = y に対する重み

フォーム f に
対応する列
(J 項目分)

y に
対応する列

BST: テスト構成モデル

- フォーム間で項目の重複を許さないための制約

$$\mathbf{I}_J = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

$$\mathbf{l}_2 = \mathbf{0}_J, \quad \mathbf{A}_2 = \begin{bmatrix} \mathbf{I}_J & \mathbf{0}_J \end{bmatrix}, \quad \mathbf{u}_2 = \mathbf{1}_J$$

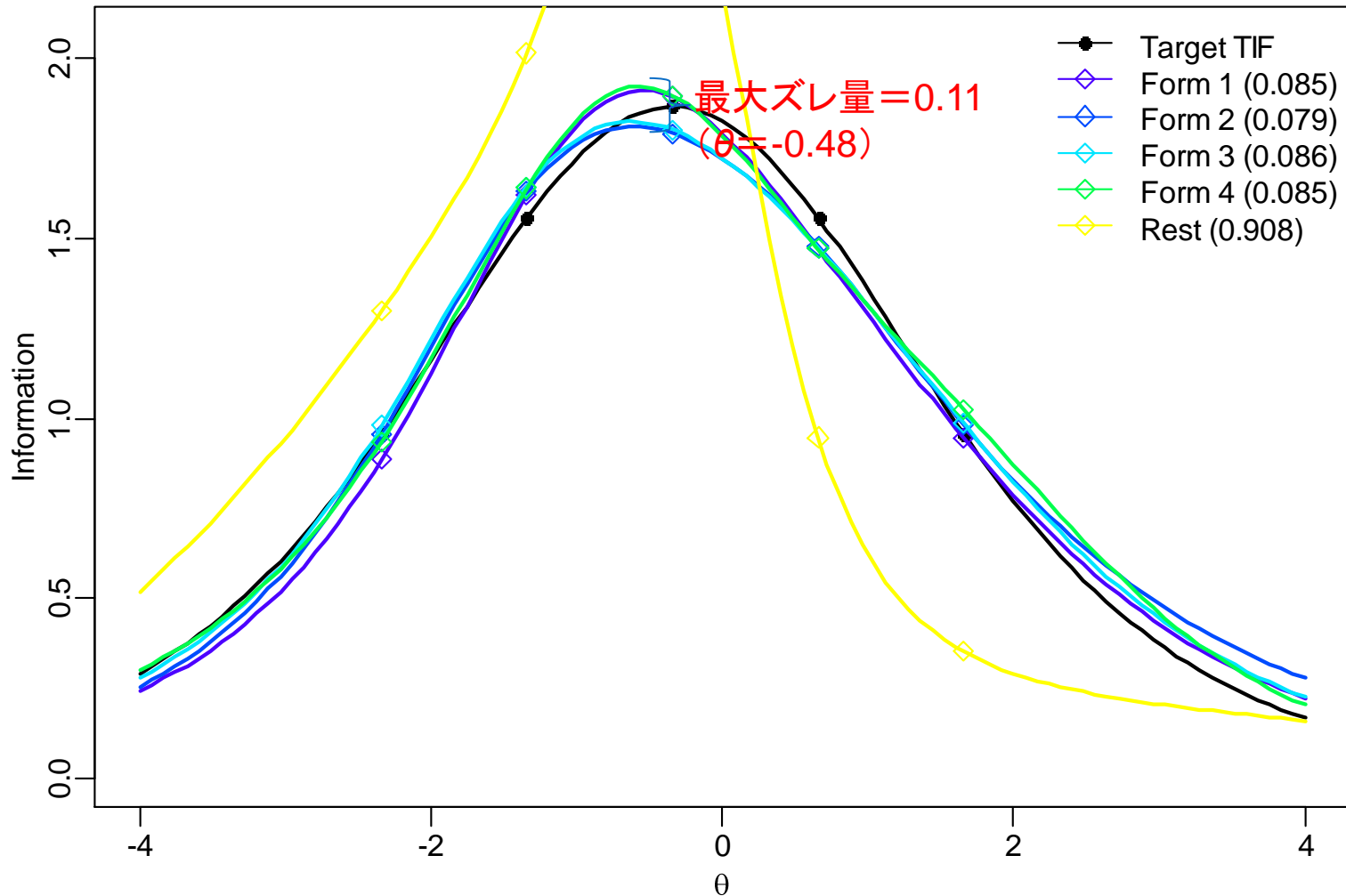
フォームに
対応する列
(J項目分)

yに
対応する列

- まとめると...

$$\mathbf{l} = \begin{bmatrix} \mathbf{l}_0 \\ \mathbf{l}_1 \\ \mathbf{l}_2 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}$$

BST: 結果



まとめ

- 同時構成 & BSTではより等質なフォームが得られた
 - ただし、同時構成は非常に時間がかかる
 - 線形計画法の枠内では、BSTが妥当な選択か
- 今回の例では、平均項目プール情報量に比較的近い目標TIFを設定
 - 両者が離れるほど、逐次構成における「劣化」はより顕著に
 - 両者が離れるほど(項目プールのサイズに対して使用する項目の割合が大きくなるほど)、同時構成／BSTでも平均的なエラーは増加→何らかの追加制約が必要

まとめ

- 項目一致数

SEQ /SIM	1	2	3	4	R
1	5	3	0	2	0
2	1	3	2	3	1
3	0	0	6	4	0
4	4	4	0	1	1
R	0	0	2	0	8

SEQ /BST	1	2	3	4	R
1	2	3	1	3	1
2	3	0	3	2	2
3	2	1	4	3	0
4	2	4	1	2	1
R	1	2	1	0	6

SIM /BST	1	2	3	4	R
1	2	2	0	5	1
2	3	2	4	1	0
3	4	2	2	2	0
4	1	3	3	2	1
R	0	1	1	0	8

全体的なまとめ

- 項目プールの維持・利用
 - 事前に, 中長期的な視点で, 綿密な計画を立てておく
 - 継続的に実行可能であることが重要・・・一旦動き始めてしまうと, 変更は困難
 - 計画段階でのパイロット研究やシミュレーション研究は非常に有用
- Input
 - 作問(領域, 項目数, 歩留まり・・・)
 - 等化デザイン(データ収集計画 & 等化法)
- Output
 - テスト仕様 & 実施計画
 - 候補項目の使用ルール
- 参考文献: Wendler & Walker (2006)

参考文献

- Belov, D. I., & Armstrong, R. D. (2006). A constraint programming approach to extract the maximum number of non-overlapping test forms. *Computational Optimization and Applications*, **33**, 319-332.
- 茨木俊秀・福島雅夫 (1993). 最適化の手法 共立出版
- Ishii, T., Songmuang, P., & Ueno, M. (2014). Maximum clique algorithm and its approximation for uniform test form assembly. *IEEE Transaction on Learning Technologies*, **7**, 83-95.
- 加藤健太郎・山田剛史・川端一光 (2014). Rによる項目反応理論 オーム社
- van der Linden, W. J. (2005). *Linear models for optimal test design*. NY: Springer.
- Songmuang, P. & Ueno, M. (2011). Bees algorithm for construction of multiple test forms in e-testing. *IEEE Transaction on Learning Technologies*, **4**, 209-221.
- Sun, K.-T., Chen, Y.-J., Tsai, S.-Y., & Cheng, C.-F. (2008). Creating IRT-based parallel test forms using the genetic algorithm method. *Applied Measurement in Education*, **21**, 141-161.
- Wendler, C. L. W., & Walker, M. E. (2006). Practical issues in designing and maintaining multiple test forms for large-scale programs. In Downing, S. M., & Haladyna, T. M. (Eds.), *Handbook of test development* (pp. 445-467), Mahwah, NJ: Lawrence Erlbaum.