IRT を用いたクラウドソーシング実験の再現性の向上手法の提案

勝野　皓太†　　松原　正樹††　　渡辺知恵美†††　　森嶋　厚行††

† 筑波大学 情報学群情報メディア創成学類　〒 305-8550 茨城県つくば市春日 1-2
†† 筑波大学 図書館情報メディア系　〒 305-8550 茨城県つくば市春日 1-2
††† 筑波技術大学産業技術学部産業情報学科　〒 305-8520 茨城県つくば市天久保 4-3-15
E-mail: †kohta.katsuno.2019b@mlab.info, ††{masaki,mori}@slis.tsukuba.ac.jp,
†††chiemi@a.tsukuba-tech.ac.jp

あらまし　クラウドソーシング実験の再現性を向上させることは，その実験結果の信頼性を高めるために必要不可欠である．しかし，クラウドソーシングに参加するワーカは不特定多数であるため再現性を向上させることは難しい．本論文ではクラウドソーシング実験の再現性を向上させるために効果的なメタデータの種類について提案する．我々はワーカがタスクを解答するために必要な能力をメタデータとして付与することが再現性を向上させる有効な手法ではないかと仮説を立てた．しかし、特定のタスクを解答するために必要なワーカの能力を特定することは難しい．この未定義の能力を見つけるために項目反応理論を適用し、ワーカの能力分布を計算できるようにした．提案手法に対して実験を行い，ワーカの能力分布が再現性の向上に有効であることがわかった．またワーカの能力を推定するために必要なコストを協調フィルタリングを用いることで削減できることを示した．

キーワード　クラウドソーシング，再現性，シミュレーション

1　はじめに

クラウドソーシング実験の再現性は，クラウドソーシングの分野における最も差し迫った懸念の 1 つである [1] [2]．再現性の向上を妨げている要因の 1 つに，不特定多数の人々がタスクに参加していることが挙げられる．同じ実験を 2 回試みても，異なるワーカ集合が参加した場合，ワーカが持つ様々な要因によって，異なる結果を生む可能性がある [3]．したがって，クラウドソーシング実験の結果が信頼できると査読者および読者を説得するために，この分野の研究者はしばしば同じ実験を何度も繰り返し，その差は統計的に有意であると主張することが一般的である．ただし，このプロセスを通じて実験結果を再現することは，時間と金銭的コストの両方の観点から容易ではない．

そこで本論文では，実験に参加したワーカに関するメタデータを実験の説明に追加することを提案する．これにより，他の研究者はメタデータを参照することで実験と同様のワーカ集合で実験を行うことができ，再現性を向上できるようになる．最初のステップとして，特定のタスクを解決するためにワーカの能力分布を使用することの有効性を調べた（図 1）．

しかし，多くの場合で，特定のタスクを解決するために必要なワーカの能力を特定することは困難である．たとえば，英語での日常会話の文章を理解するワーカの能力は，TOEFL などで測ることができる．しかし，一般的にタスクを解決するワーカの能力は，特定のタスクに依存し，上記の例のような既知の問題でない限り，明確な答えはない．

ここでは，項目反応理論（IRT:Item Response Theory）[4] を適用してこの未知の能力を見つけ，ワーカの能力分布を示す．IRT は，TOEFL などの標準化されたテストで広く使用されている．IRT を用いることで，タスクを解くワーカの能力というものを θ と定義して推測することができる．また IRT は，ワーカの能力を見つけるだけでなく，ワーカの能力を測定するための優れたテストを作ることができる．

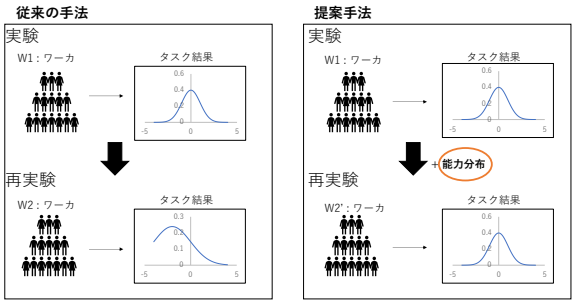

図 1　本稿ではワーカの能力分布に着目しクラウドソーシング実験の再現性を向上させることが目的である．

本論文の構成は次の通りである．まず，第 2 章で関連研究について述べる．第 3 章では，本論文で使用する IRT についての予備知識について述べる．第 4 章では，IRT によって推測されたワーカの能力分布を用いたクラウドソーシング実験の再現性を向上させる提案手法について述べる．第 5 章では，第 4 章で提案した手法を用いてシミュレーションを行い，評価をする．第 6 章では，ワーカの能力を推定するテストを作るためのコスト削減をこなう提案手法について述べる．第 7 章では，第 6 章で提案した手法を用いてシミュレーションを行い，評価をする．第 8 章では，実験全体の考察について述べる．第 9 章では，まとめと今後の課題について述べる．

2 関 連 研 究

ReproZip [5]，Reprowd [1] は実験で使用したプログラムコードを実験結果とともにパッケージとして共有するツールである．これらの提案手法では実験結果とともに実験を行ったプログラム環境全てを他者と共有することができ簡単に再現実験を行えるように促している．本研究の目的は再現性を向上させる点にある為異なる．

Rehab らはクラウドソーシングの再現性とプラットフォームの関係について研究した [6]．この研究によると異なるプラットフォーム間で再現実験を行うとタスク結果の品質水準が大幅に異なるが，同じプラットフォーム内で再現実験を行えばタスク結果の品質は安定する．本研究は同じプラットフォーム内での再現性の向上を目指しており異なるプラットフォーム間での再現性は考慮していない為今後の課題となる．

Williams らはワーカが同じタスクを 2 回繰り返した時のタスク結果の一貫性について研究した [7]．この研究ではタスクキューの中にタスクを複製し重複しタスクを一貫して行ったワーカの信頼性を調べた．本研究の目的では異なるワーカに同じタスクを解いても再現性を向上させることである為新規性があると言える．

3 IRT に関する予備知識

IRT とはテスト結果の分析によく用いられる理論で，各タスクと各ワーカは独立していると仮定され，ワーカのタスク結果からそれぞれのタスクの困難度やワーカの能力を推定できる手法である [4]．IRT には多様な確率モデルがあるが今回は一番使われている 2 パラメータモデル (2PL) を使用する．

2PL でタスク j に正答する確率は以下の式で示される．

$$P_j(\theta) = \frac{1}{1 + exp(-Da_j(\theta - b_j))}, -\infty < \theta < \infty$$

θ は能力パラメタと呼ばれ各ワーカの能力を表すパラメータである．a と b は各タスクの特徴を表すパラメータであり a を項目識別力，b を項目困難度と呼ぶ．項目識別力とは項目がどれだけ低い能力のワーカと高い能力のワーカを区別できるかを表し，値が大きいほど能力の低いワーカと能力の高いワーカを区別できるということである．項目困難度とは項目の難しさを表している．一般に b の値が高いほどその項目は難しいということになり，正答確率は低くなる．D は尺度因子と呼ばれるものであり，一般に 1.7 という値が用いられる．

4 ワーカの能力分布に基づいたクラウドソーシング実験の再現性の向上

ワーカの能力分布に着目して，クラウドソーシング実験の再現性を向上する手法を提案する（図 1）．従来の方法では，実験に参加したワーカ（$W1$）と再実験に参加したワーカ（$W2$）は，θ に関して異なる集合であることが多い． したがって，ワーカの能力分布は異なる可能性があり，タスクの結果も異なっていくため実験を再現することは難しかった．

しかし，提案手法では，実験者はタスクの結果に加えてワーカの能力分布，またワーカの能力分布を求めるために用いたテストの 2 つをメタデータとして報告する必要がある．この 2 つのメタデータがあることにより，別の研究者が実験を再現しようとする場合，メタデータとして添付されたテストを使用してワーカの能力を推定することができる．したがって $W1$ に類似した分布を持つワーカ $W2' \subseteq W2$ を選択することができ，タスク結果の再現性を改善できると考えた．

4.1 実 験 概 要

実験の再現性に対するワーカの能力分布の影響を求めるために，実験 1 を実施する．実験 1 では過去に行ったクラウドソーシング実験のタスク結果を用意し，参加したワーカ全員の能力分布と等しくなるように一部のワーカを選定しタスク結果が再現するか確認するシミュレーションを行う．

そのためにまずタスク結果から IRT を適用しすべてのタスクの項目識別力，項目困難度を推定し，テストを作成する．テストを作成するとワーカの能力を求められるようになるため，IRT でワーカの能力を推定し能力分布が等しくなるようにワーカを選定する．タスク結果の再現性を提案手法を用いない場合と比較するために，能力分布が等しいワーカ集合 $W2'$ と，能力に関係なく無作為に選んだワーカ集合 $W2$ を用意する．ここで能力分布が等しいワーカ集合を `Selected`，無作為に選んだワーカ集合を `Random` とする．2 つの集合で，母集合である実験に参加したすべてのワーカのタスク結果の正答率の平均値と 2 つの集合内でのタスク結果の正答率の平均値との差を求める作業を 10,000 回繰り返し，標準偏差を求める再現性の基準として標準偏差から区間推定を行い有意差が得られるまで必要な試行回数の割合で比較する．

4.2 テストの作り方

実験 1 で用いるテストは次のように作成する．第 3 章で述べたように項目識別力 a が高いタスクは，そのタスクの項目困難度 b 付近のワーカの能力をより正しく識別できる．そのため初めにタスク結果から IRT を適用して得られたパラメータ（a, b）から a の値が高いタスクを選ぶ．その中から能力の低いワーカから高いワーカまで識別できるように b の値が能力の定義域を埋め尽くすようにタスクを選ぶ．

5 実 験 1

5.1 データセット

この実験では，現実世界のワーカから取得した一連のマイクロタスクの結果を使用した（詳細は [8] にある）． このタスクは掲示された絵画を見て，異なる 4 人の画家の絵画 4 枚と比較しどの画家の作品であるかを当てる多肢選択タスクであり，実際のタスク画面を図 2 に示す．実験に参加した 85 人のワーカが 96 枚の絵画の画家を識別するように求められた．そのうち選択肢に使われている 4 つの絵画を除いた 92 枚の絵画をシミュレーションに利用した．

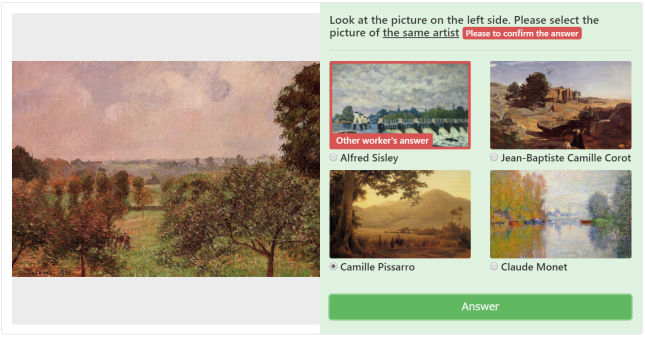

図 2　タスクの画面．ワーカは左の絵画を見てその作者を右の 4 つの選択肢から選ぶタスク．

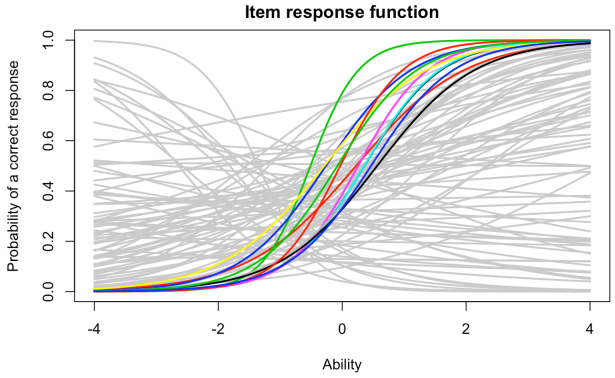

図 3　実験 1 で使用するテストの項目特性曲線．色が付いているものは実際にテストとして選んだタスク．灰色の曲線はそれ以外のタスク．

Selected，Random は 85 人のワーカから 20 人を選ぶこととする．Selected は 85 人のワーカの能力分布 θ の平均値から 0.01 以内，標準偏差は 0.05 以内を条件とし，Random は能力に関係なく無作為に選んだ．

5.2　テスト作成

データセットに IRT を適用した項目特性曲線を図 3 に示す。今回のデータセットでは傾きが負である曲線や、平らな曲線などを描くようなテストに相応しくないタスクが混ざっていた。また項目識別力が高いタスクを選んでみると項目困難度に差がなく能力 0 付近に集まっていることがわかる。テストは本来、項目識別力が高く項目困難度が高いものから低いものまであることが望ましい。しかし、今回は項目困難度に差がなかったため、項目識別力が高い 12 個のタスクを選んだ。

5.3　結　　果

予備実験の結果は (Table 1) に示す．棒グラフが平均誤差の平均を表していて，線が標準偏差を表している．標準偏差は集合 (Selected) が 0.5719585，集合 (Random) が 0.9971918 になった．以上より標準偏差は集合 (Random) と比べて約 40.4% 削減することができた．

ここで得られた Selected と Random の標準偏差から区間推定を行うことで再実験に必要な試行回数の割合を求めた．区間推定を行うためには標準偏差，許容誤差，信頼度が必要である．

表 1　標準偏差の結果 (SD)

	Selected	Random
SD	0.57	1.00

サンプルサイズ n，標準偏差 σ，信頼度 95%，および許容誤差 δ が与えられた場合，$1.96 \times \frac{\sigma}{\sqrt{n}} = \delta$ という式が得られる．そのとき再実験に必要な試行回数の割合は下記の式になる．

$$ratio = \frac{n_{selected}}{n_{random}} = (\frac{\sigma_{selected}}{\sigma_{random}})^2$$

その結果，Selected グループで必要な再実験に必要な試行回数は Random グループの約 32.9 %であった．したがって，Random のワーカーを使用した場合と比較して，タスクの数（および試行回数）を 1/3 に減らすことができる．これは 1/3 の金銭と時間のコストで再実験を行えることを意味する．

5.4　考　　察

表 1 で Selected の標準偏差は Random の標準偏差と比べて小さかったため再現性は向上したと言える．実験 1 で得られた結果より，IRT を用いて推測したワーカの能力分布をクラウドソーシング実験のメタデータとして使うことの有用性が得られ，再現性を向上するために有効であることがわかった．

また提案手法ではワーカ間のタスク結果に差があればあるほど効果的である．なぜなら差が小さいデータセットでは Random ワーカの能力分布が再現したい実験のワーカの能力分布と相似し，Selected と変わらない結果になるからである。その意味で，今回用いたデータセットのタスク結果の標準偏差は 0.09 であり，あまり提案手法による効果が発揮できなかったと言える．そのため能力の高い人は解け，低い人は解けないタスクを用意することで，タスク結果に差が生まれより再現性が向上されると考えている．

6　ワーカ人数の削減手法

6.1　テスト作成に必要なタスク解答数の削減手法

実験 1 よりテストを用いて推定したワーカの能力分布を用いることで実験に必要な試行回数を減らせることがわかった．しかしいくらサンプルサイズを減らせても，ワーカの能力を測るテストの作成コストが高いと提案手法は実用的ではない．したがって本章ではより少ないコストで正しく能力を測れるテストを作る手法を提案する（図 4）．

IRT を適用するためにはすべてのワーカがすべてのタスクを解いたデータが必要であり，従来手法では 6 人のワーカが必要な時，6 人全てに全てのタスクを解答させていた．しかし，全てのタスクを解いてもらうことはワーカにとって大きな負担であり研究者にとってもコストがかかるものであった．

提案手法では 1 人で全てのタスクを解くのではなく，複数のワーカが数タスク解答しそのタスク結果を組み合わせて 1 人分の解答を作成する．図 4 では仮にワーカ 3 人分の解答を 6 人で作るとする．必要なタスク解答数は 3 人 ×6 問より 18 データとなる．これを 6 人でタスクを解答するので 1 人当たり 3 問解

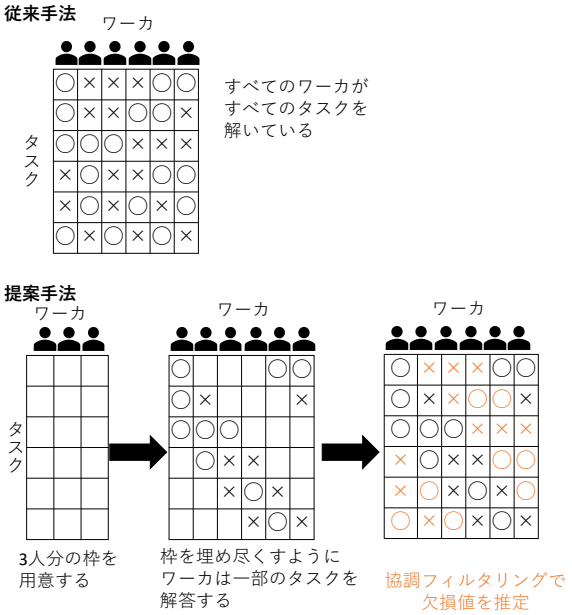

図 4　テスト作成に必要なワーカ人数削減手法の概要．ワーカはタスクの一部だけを解き欠損値を協調フィルタリングで推定することで 6 人必要だったコストが半分になる．

答すれば良いことになる．従来手法では 1 人当たり 6 問のタスクを解いていたが提案手法では 3 問のタスクだけ解答すれば良いためコストを 1/2 に抑えることができる．

しかし，テストを作るには本来 6 人のワーカが必要であったためこのままでは 3 人分のタスク結果が足りない状況になる．ここで用いる手法が協調フィルタリングである．協調フィルタリングを用いることで欠損値を推測することができる．6 人がそれぞれ解いた 3 問のタスク結果から残りの欠損値を協調フィルタリングを用いて推測し，3 人分のタスク結果から 6 人分のタスク結果を作成する．

実験 2 では 1 つのタスクにつき何人が解けばテストを作成できるか調べる．

6.2　協調フィルタリング

協調フィルタリングとは，あるワーカと類似したワーカを見つけ，類似したワーカ間では解答パターンが相似するといった仮定で欠損値を補完する手法である [9] [10]．協調フィルタリングは推薦アルゴリズムとも呼ばれ，Amazon などのレコメンデーションなどに使われている．協調フィルタリングには様々な手法が存在するが今回は有名な Matrix Factorization という手法を用いる．

6.2.1　Matrix Factorization

Matrix Factorization（以下，MF と呼ぶ）とは各ユーザが解いたタスクの解答パターンを行列とし，その値からワーカの特徴を持つ行列とタスクの特徴を持つ行列の二つに行列分解する手法である [11]（図 5）．通常の IRT では正解を 1 不正解を 0 として計算を行うが，MF では推測したい欠損値を 0 としておくので正解を 5 不正解を 1 として MF を適用し得られた結果の 3 以上を正解，3 未満を不正解として扱うことにした．

6.2.2　Score Based Prediction

また比較対象として MF のほかにワーカの得点に注目し近い得点同士のワーカで解答パターンを共有する手法を提案する．（以下 SBP : Score Based Prediction と呼ぶ.）この手法は得点が近いワーカ間で解答パターンは相似するといった仮定をもとに推測する手法である．最初に同得点のユーザのみで比較し，次に ±1 で比較，±2 で比較とすべての欠損値を補完するまで範囲を広げていくアルゴリズムである．また 0 と 1 が同時に存在していた場合はそれぞれの割合の確率でランダムに選ぶことにした．図 6 で概要を説明する．まず図中の 1 はそのタスクを正解したこと，0 は不正解，それ以外は無解答を表している．これに SBP を用いるとワーカの得点はそれぞれ 2 点，1 点，2 点になる．従って得点が同じであるワーカ 1，3 に着目しワーカ 1 のタスク 3，ワーカ 3 のタスク 2 が補完される．またワーカ 2 は他に同得点のワーカがいないため 1 回目のループは終了する．次に ±1 の範囲でワーカを見るとすべてのワーカが当てはまるのでタスク結果を比較し補完する．ここですべての欠損値が補完されたためアルゴリズムが終了する．

実験 2 ではテスト作成に必要なタスク結果量と 2 つの手法について比較するためにシミュレーションを行って調査する．

	i_1	i_2	i_3	i_4
u_1	5	5	0	1
u_2	5	0	5	0
u_3	1	0	0	5

→

	i_1	i_2	i_3	i_4
u_1	5	5	2.4	1
u_2	5	4.9	5	5.1
u_3	1	0.9	3.4	5

図 5　Matrix Factorization を適用したときの例

タスク

ワーカ	t_1	t_2	t_3	t_4
w_1	1	1	-	-
w_2	-	0	-	1
w_3	1	-	1	-

→

	t_1	t_2	t_3	t_4
w_1	1	1	1	1
w_2	1	0	1	1
w_3	1	1	1	1

図 6　Score Based Prediction を適用したときの例

7　実　験　2

7.1　データセット

実験 2 では実験 1 で使用した 85 人のワーカが 92 問のタスクを解いたデータを利用する．

今回は一つのタスクにつき何人が解答すればテストを作れるかを調べたい。そのため 10 人から 70 人まで 10 人刻みでワーカ人数を増やしていった 7 パターンを作り，それぞれに該当する必要なタスク数になるまで欠損値を作成する．例えば 10 人分のタスク結果を用意する場合は全 7820 データ中 920 データを残してそれ以外を欠損値とする．

また欠損値の作り方に影響が出ないように全てのタスクが同じ割合で解答されるようにワーカ 1 がタスク 1 からタスク 10 を解きワーカ 2 がタスク 11 からタスク 20 を解くといった階段上になるパターン round-robin と，ランダムにタスクを選

ぶパターン random を用意した．そのため実験 2 では round-robin_SBPround-robin_MF，random_SBP，random_MF の 4 つの手法を欠損値の割合 7 パターンで実験を行う．

7.2 比較方法

比較方法として，本来のデータセットから得られた各ワーカ i の能力値 $\theta_{85}(i)$ と欠損値の割合が a 人のときの推定した能力 $\theta_a(i)$ との絶対値の差の平均について比べる．式は以下の通りである．

$$\frac{\sum_i |\theta_{85}(i) - \theta_a(i)|}{85}$$

また上記に加えて実験 1 と同様に 10 人，40 人，70 人でサンプルサイズを計算し比較した．

7.3 結果

実験で得られたワーカの能力を（図 7）に示す．横軸は欠損値の割合で用いた仮のワーカの人数を表しており，値が大きいほど欠損値は少ない．縦軸は能力の差の平均を示している．

得られたワーカの能力分布から計算した再実験に必要なサンプルサイズは（図 8）になる．縦軸は再現するために必要な試行回数の割合を示している．点線は実験 1 で得られた 32.9%を表す．

7.4 考察

図 7 についてみていくと全ての手法で仮想ワーカの数が小さいときには 4 つの手法で差があまりないことがわかる．しかし仮想ワーカの数が増えていくにつれて SBP は MF と比べてより能力の差が小さくなったと言える．欠損値の取り方に着目すると SBP，MF についてそれぞれあまり影響が出ないことがわかる．

次に図 8 についてみていく．実験 1 で得られたサンプルサイズの結果である 32.9%と比較すると 40 人の時の random_SBP，70 人の時の round-robin_SBP の 2 つが 32.9%を下回る結果となった．したがって random_SBP を用いることで 85 人と比べて約 50%のワーカコスト削減が行えた．

また MF は 10 人，40 人とサンプルサイズが高かったが 70 人では SBP と近い値を出すことができた．これは能力に差があっても等しい能力分布を取り出すことができれば再現することができる実験 1 の結果を改めて確約するものとなった．

8 全体の考察

ワーカの能力分布を IRT で推定しメタデータとして用いることでクラウドソーシング実験の再現性は向上することがわかった．これは同様のワーカの能力分布間でワーカのタスク結果が再現するという仮定が正しかったと言える．

実験 2 ではテストを作成するコスト削減に random_SBP を用いることでワーカの人数が 10 人であっても 80%まで削減できることがわかった．また SBP と MF を比較すると全体を通して MF の結果が悪かったと言える．これは協調フィルタリングの特徴として，似た回答パターンを持つワーカは同じ回答になるというものがある．これによりすべてのタスクで全く同じ

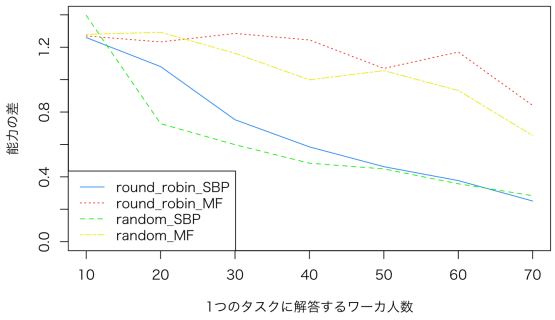

図 7 基となるワーカ全員の能力と実験で推定したワーカの能力値との絶対値差の平均

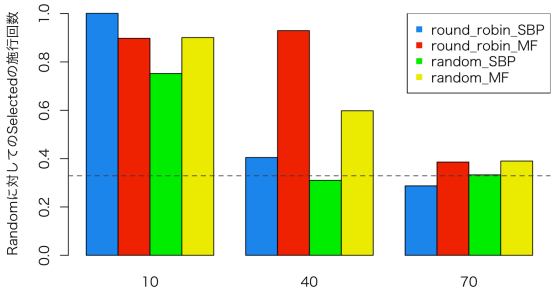

図 8 実験 1 と同様に Selected と Random を計算し得られたサンプルサイズ

回答を持つワーカの数が SBP と比べて圧倒的に多かった．そのためワーカの能力に差が生まれにくく，`Selected` と `Random` にも差があまり生まれなかったからだと考えられる．

今後の課題として実験 1 の考察でも述べたように，よりワーカのタスク結果に差が生まれるよう偶然正解しないタスクを用意するべきと考えられる．IRT の考え方として，能力 1 の人は困難度 1 以下の問題はすべて正解し，1 よりも難しいものはすべて間違えるという仮定がある．つまり，自分の能力以上の問題に偶然正解してしまうと IRT の結果が悪くなるというものである．今回のデータセットではワーカは 4 択の問題を取り組んでもらった．しかし，4 択では分からなくても 25%の確率で正解してしまう．そのため選択肢を増やすことで本当にわかる問題だけ正解するようなタスク設計を作成することが必要である．

またタスクの難易度調整も大事であり，あまりに難しすぎたり簡単すぎたりするとタスク結果が偏り，能力に差が生まれなくなる．今回のタスクは 92 問あり平均正答率は 35.6 問とやや難しいタスクであったと言える．そのため，選択肢と難易度の 2 点を変更することで再現性の結果はまだ向上できると考えている．

実験 2 のワーカ人数の削減ではワーカが 10 人で 80%までコスト削減することができた．しかし，実際のクラウドソーシング実験では 1 つのタスクに 10 人解くようなタスクは非常に稀である．実用段階まで提案手法を上げるには，3 から 5 人程度までワーカ人数を削減したい．そのために SBP ではワーカの

得点だけを見てタスク結果を予測していたが，得点が同じワーカの中から回答パターンを見てより細かい集合を作り，その中でタスク結果を予測することで SBP の精度を上げていく必要がある．

また本論文ではシミュレーションで再現性の向上を確認したため，今後は実際にクラウドソーシング実験を行って実データで再現性が向上されるか調査する必要がある．

9 結　　論

本論文では，IRT を用いたワーカの能力分布をメタデータとする手法を提案し，クラウドソーシング実験の再現性が向上されるかシミュレーションによる調査を行った．シミュレーションの結果よりワーカの能力分布をメタデータとして用いることで再実験に必要な試行回数が約 1/3 のコストまで抑えることを示した．また能力を測るために必要なテストを作成するコストをワーカ人数を 10 人まで減らしても，試行回数を 80%までに抑えることができた．

今後の課題として実際にクラウドソーシング実験を行い，実データ上で再現性が向上されるかを調査する必要がある．またテスト作成におけるコスト削減において新しい協調フィルタリングを用いてさらにコスト削減する必要がある．

謝　　辞

本研究の一部は JST CREST (JPMJCR16E3)，JST AIP チャレンジ，JSPS 科研費 (JP19K11978) の助成を受けたものです．

文　　献

[1] Ruochen Jiang and Jiannan Wang. Reprowd: Crowdsourced data processing made reproducible. *CoRR*, Vol. abs/1609.00791, , 2016.

[2] Praveen Paritosh. Human computation must be reproducible. In *WWW 2012, Lyon.*, 2012.

[3] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Comput. Surv.*, Vol. 51, No. 1, pp. 7:1–7:40, January 2018.

[4] Kim S.H. (Ed.) Baker, F. (Ed.). *Item Response Theory: Parameter Estimation Techniques*. Boca Raton: CRC Press, 2004.

[5] Fernando Chirigati, Rémi Rampin, Dennis Shasha, and Juliana Freire. Reprozip: Computational reproducibility with ease. In *SIGMOD 2016 - Proceedings of the 2016 International Conference on Management of Data*, Vol. 26-June-2016, pp. 2085–2088. Association for Computing Machinery, 6 2016.

[6] Gianluca Demartini Rehab Qarout, Alessandro Checco and Kalina Bontcheva. Platform-related factors in repeatability and reproducibility of crowdsourcing tasks. In *Proceedings of the Seventh AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2019, WA, USA, October28-30, 2019*, pp. 135–143, 2019.

[7] Williams. Alex C., Joslin Goh, Charlie G. Willis, Aaron M. Ellison, James H. Brusuelas, Charles C. Davis, and Edith Law. Deja vu: Characterizing worker reliability using task consistency. In *Association for the Advancement of Artificial IntelligenceI Conference on Human Computation and Crowdsourcing*, Québec City, Québec, Canada, 2017. The AAAI Press, Palo Alto, California, The AAAI Press, Palo Alto, California.

[8] Masaki Kobayashi, Hiromi Morita, Masaki Matsubara, Nobuyuki Shimizu, and Atsuyuki Morishima. An empirical study on short-and long-term effects of self-correction in crowdsourced microtasks. In *AAAI HCOMP 2018*, 2018.

[9] 土方嘉徳. 嗜好抽出と情報推薦技術. 情報処理, Vol. 48, No. 9, pp. 959–961, 2007.

[10] Ristu Saptono. User-item based collaborative filtering for improved recommendation. 03 2010.

[11] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, Vol. 42, No. 8, pp. 30–37, Aug 2009.