DEIM Forum 2022 E41-4

タスク結果の品質と労働負荷分散を両立したタスク割当て手法

根岸　寛太†　　伊藤　寛祥††　　松原　正樹††　　森嶋　厚行††

† 筑波大学 知識情報・図書館学類　〒 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
E-mail: †kanta.negishi.2021b@mlab.info, ††{ito,masaki,mori}@slis.tsukuba.ac.jp

あらまし　クラウドソーシングプラットフォームにおいて，リクエスタは期待される品質のタスク結果を得るためにワーカが過去に実施したタスクの評価結果，またはテストの成績をもとにワーカを選択する．リクエスタが高品質のタスク結果を得るための方法の一つはスキルレベルが要求水準よりも高いワーカのみにタスクを割当てることである．しかし，スキルレベルが要求水準を満たさないワーカを候補から除外するというワーカ選択方法はワーカの雇用機会への制限や，リクエスタが利用可能な労働力の不足などの問題を引き起こす．この論文では，ワーカのスキルレベルとタスクの難易度の推定に基づき，できるだけ多くのワーカ間でタスクを分担するための手法を提案する．提案手法では，タスク結果の品質を要求水準以上に満たしつつ，ワーカ間の担当タスク数の分散が最小となるようにタスクを割当てる．実験の結果，タスクに取り組むことができるワーカの人数を増やすことができたが，タスク結果の品質維持における課題が明らかとなった．

キーワード　クラウドソーシング，データマイニング

1　はじめに

近年ではクラウドソーシングプラットフォームが普及し，多くの人々がワーカとしてタスクに取り組んでいる．クラウドソーシングプラットフォームでは，仕事の依頼主 (リクエスタ) がタスクと呼ばれる仕事を発注し，働き手としてプラットフォームに登録している人々(ワーカ) にタスクが割当てられる．タスクの割当てとは，どのタスクをどのワーカに取り組ませるかというマッチングを決定することである． 本論文ではマイクロタスクのタスク割当てを取り扱う．マイクロタスクとは数秒から数分で開始できる単純な仕事であり，一般にリクエスタとワーカの間に指示文以外のコミュニケーションはないという特徴がある [1,2]．例えば，データ収集やアノテーションなどの作業がマイクロタスクとしてワーカに依頼されている．

必ずしもワーカについて十分な情報が得られるとは限らないマイクロタスクの割当てにおいて，リクエスタがタスクの結果を高品質に保つためにはワーカの選択が重要な工程である．一般的にリクエスタが用いる品質管理手法の一つはワーカを選択する際，ワーカが過去に取り組んだタスクや能力テストの正解率などを参照して適当なワーカを選択することであり，結果としてリクエスタは能力のレベルが要求水準 (threshold) よりも高いワーカを選択することになる．例えば，代表的なクラウドソーシングプラットフォームの 1 つである Amazon Mechanical Turk[1]では，リクエスタは過去のタスクの承認率が threshold よりも高いワーカに限定してタスクを割当てることが可能であり，この機能によってリクエスタは安定して高品質のタスク結果を得ることができる．しかし，threshold のみを基準にしたワーカの選択方法は割当て候補のワーカの数を制限し，リクエスタとワーカの両方に負の影響を及ぼすことがある．具体的には，リクエスタの観点では労働力の不足やタスク完了の遅延などの問題が起きる可能性がある．また，ワーカの観点ではタスクを与えられたワーカの労働負荷が大きくなる一方で，実際にはタスクに取り組むために十分なスキルを持っているワーカが過去のタスクの成績が高くないために割当ての候補から除外されてしまう可能性がある．

そこで，本研究ではタスク結果の品質を維持しながら，様々なレベルのスキルを持つワーカ間でタスクを分担させるようなタスク割当て手法を実現することを目的とする．ここではタスク結果の品質をタスクの正解率，労働負荷を各ワーカに割当てられるタスクの数，そしてワーカのスキルをタスクに正解する能力を表すものと定義する．タスクをワーカに割当てる際，結果の品質とタスクの分担を両立することは簡単ではない．単純に高い結果の品質を得るためには図 1 のようにタスクの困難度は考慮せず，高いレベルのスキルを持つワーカのみにタスクを割当てればよいが，この方法ではワーカ間でタスクを分担することができない．つまり，タスクを与えられたワーカの労働負荷が大きくなる．また，単にタスクの分担を優先する場合は図 2 のようにランダムに割当てればよいが，ワーカのスキルを割当てに考慮していないためタスク結果の品質が下がってしまう．そこで，本研究は図 3 に示すように，異なるスキルを持つワーカのそれぞれのスキルレベルに合った難しさのタスクを割当てることで，結果の品質を維持しながらワーカ間でタスクを分担させることを目指す．本論文のリサーチクエスチョンは以下の二つである．

RQ1: ワーカのスキルとタスク難易度を考慮する割当てによってどれほど品質と労働負荷分散を両立できるか？

RQ2: 能力とタスク難易度に基づきタスクを割当てる実用的なアルゴリズムは存在するか？

1：https://www.mturk.com/

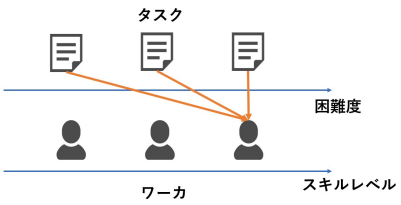

図 1　スキル上位のワーカへ割当て

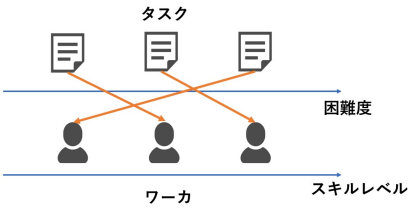

図 2　ランダムな割当て

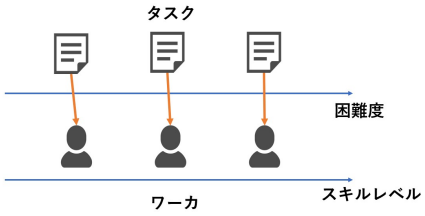

図 3　スキルレベルによる割当て

この論文ではワーカの能力とタスクに必要なスキルの推定に基づいて，より多くのワーカに仕事の機会を与えるための手法を提案する．提案手法は IRT(Item ResponseTheory) [3] モデルに基づく．IRT とは，タスクの困難度とワーカの能力の分布を求めるための理論である．IRT は TOEFL などの標準化されたテストの作成に広く用いられてきた．本研究ではワーカの能力を表すパラメータ θ を求めるために IRT を利用する．本研究の提案手法は以下のアイデアに基づく．ワーカの集合 $\mathcal{W}$ とタスクの集合 $\mathcal{T}$ が与えられたとき，タスクの補集合 $\mathcal{T}' \subset \mathcal{T}$ を選択して $\mathcal{W}$ 中のワーカの能力を推定するためのテストを作成する． そして，$\mathcal{T} - \mathcal{T}'$ のタスクについて，そのタスクを正しく完了できると推定されたワーカに割当てる．ここで，問題はまだどのワーカも回答していない $\mathcal{T} - \mathcal{T}'$ のタスクの困難度をどのように推定するかということである．本研究の提案手法では，タスクに対してその回答と確信度の値を出力する AI モデルを導入し，その出力をまだワーカが取り組んでいないタスクの困難度を推定するために用いる．

提案手法の有効性を検証するために，AI のタスク回答を修正するタスクにワーカが取り組むというシナリオのもと，シミュレーション実験によって割当てを評価する．この研究の貢献は以下の通りである:

- IRT に基づいたタスク割当アルゴリズムを提案し，結果品質をある程度維持しながらワーカがタスクに取り組むことのできる機会を増加可能であることを示した．
- AI モデルにおける困難度と人間にとってのタスクの難易度が大きく異なる場合があることを示し，そのような場合の課題について議論した．

本論文の構成は次の通りである．まず第 2 章でタスク割当てに関する関連研究を整理する． 次に第 3 章では本研究の取り組む問題の定義と提案手法について説明する．第 4 章では評価実験の内容とその結果について述べ，第 5 章で提案手法と評価実験の結果を考察する．最後に第 6 章で本研究のまとめと今後取り組むべき課題について述べる．

2　関 連 研 究

本章では，タスク割当てに関する関連研究，特にワーカの参加率を考慮したタスク割当て手法，またワーカのスキルを考慮したタスク割当て手法と本研究の関係について述べる．

2.1　ワーカの参加率を向上するためのタスク割当ての研究

橋本ら [4] は様々なタスクで構成される複雑なワークフローにおいて生産性，スループットといったリクエスタ視点の有効性と，タスクへの参加率や担当タスク数の標準偏差などのワーカ間でタスクを分担するための指標を両立する割当てを見つけることを目的としている．橋本らの研究によって，ワーカのタスクへの参加率と担当タスク数の標準偏差を優先した割当て戦略によってリークエスタとワーカの双方にとっての有効性が両立できることが示されている．また，Huang ら [5] は動画の字幕作成ワークフローにおいて，分解されたマイクロタスクに異なる言語スキルを持つワーカをできるだけ参加させるための手法を提案し，スキルが相対的に低いワーカもタスクの一部に参加させるための知見を示している．これらの研究はより多くのワーカによるタスクへの参加や分担を目的に含めているという点で本研究と共通している．しかし，これらの研究はワーカの所有するスキルとタスクの要求するスキルがあらかじめ入力として与えられる状況を想定しており，新しく与えられたワーカや，現実に存在する多様な種類のタスクの割当てに対応することができない．

また，空間クラウドソーシングの文脈で social fairness としてワーカ間の労働負荷の分散を最小化すること [6] を目的とした手法やワーカ間で獲得報酬を平等に分け合う手法 [7] が提案されている．しかし，これらの手法はタスクの結果品質に大きく影響する要因であるワーカのスキルを考慮していない．本研究はワーカが持つスキルの品質への影響が大きいマイクロタスクの割当てを取り扱うため IRT によってワーカの能力を推定する手法を提案する．

2.2　ワーカのスキルを考慮したタスク割当ての研究

タスクの結果品質管理のためにワーカのスキルを評価する手法が提案されている． Fan ら [8] はタスク間の類似度と，過去のワーカの成績をもとにまだワーカが取り組んでいないタスクについてワーカの正解する能力を推定する手法を提案している．具体的には，タスクの内容をもとにタスク間の類似度のグラフを作成し，ワーカのテストタスクの成績をプロファイルとして personalized pagerank を計算している．この手法によって割当て結果の正解率とスループットを向上できることが示されている．本論文の手法では IRT を用いてワーカのを推定する．

また，Duan ら [9] はタスクを構造化した上でワーカが得意

なタスクを割当てるという手法を提案している．Hettiachchi らの研究 [10] では認知能力のテストを用いてワーカの様々なタスクのパフォーマンスを推定する手法を使って分類，数え上げ，文字起こし，感情分析などのタスクについてワーカのスキルを推定したうえで割当て，ワーカの正解率を向上することができている．上記の研究はタスク割当ての前にワーカの所有するスキルやタスクに必要な知識，能力を推定しているという点で本研究の手法と共通点を持つ．

しかし，これらの研究の目的関数は主に結果の品質やタスク割当て効率のスループットなどのリクエスタの利益のみを重視した指標であり， より多くのワーカがタスクに取り組めるようにワーカ間でタスクを分担するという目的は目指されていない．

以上の関連研究を踏まえ，この研究では過去のタスク結果からワーカの能力とタスクの特徴を推定し，全体のタスク結果品質を損なうことなくワーカ間の労働負荷分散を最小化するための手法を提案する．

3 提 案 手 法

本研究で対象とする問題はタスク結果の品質を損なうことなく，できるだけ多くのワーカ間でタスクを分担させることである．この章では問題の定義と提案手法のアルゴリズムについて説明する．

3.1 問 題 定 義

$\mathcal{W} = \{w_1, w_2, ..., w_n\}$ をワーカ集合，$\mathcal{T} = \{t_1, t_2, ..., t_m\}$ をタスク集合とする．各タスク t_i はタスク識別子 i とタスクの特徴 $d_i \in \mathcal{D}$ から構成され，$t_i = (i, d_i)$ のように記述される．

ここで，次のように問題を定義する: $\mathcal{W}$ と $\mathcal{T}$ そして結果品質の threshold Th が与えられたとき，以下の (1)，(2) の要求を満たすワーカとタスクの割当てのタプル集合 $\mathcal{A}_k = \{(w_{k_1}, t_{k_1}), (w_{k_2}, t_{k_2}) \ldots, (w_{k_m}, t_{k_m})\} \subset \mathcal{W} \times \mathcal{T}$ を求める．

(1) タスク結果の品質を threshold 以上に保つ．

$$\frac{1}{|\mathcal{A}_k|} \sum_{i=1}^{|\mathcal{A}_k|} \delta(ans(w_{k_i}, t_{k_i}) = \hat{ans}_{k_i}) \geqq Th, \tag{3.1}$$

式 (3.1) 中の $ans(w_{k_i}, t_{k_j})$ はワーカ w_{k_i} の タスク t_{k_j} への回答で，$\hat{ans}_{k_j}$ は t_{k_j} の ground truth である．ここで品質はワーカに割当てたタスクの正解率を表すものとする．

(2) ワーカ間のタスク割当て数の分散 $d(A_k)$ を最小化する．各ワーカに割当てられるタスクの数のばらつきを最小化することで，ワーカ間の労働負荷を分散させることを目指す．

$$\min_{\mathcal{A}_k \subset \mathcal{W} \times \mathcal{T}} d(\mathcal{A}_k), \tag{3.2}$$

$d(A_k)$ は次のように定義される．

$$d(A_k) = \frac{1}{|\mathcal{W}|} \sum_{i=1}^{|\mathcal{W}|} (|\sigma_{w_i}(\mathcal{A}_k)| - \overline{|\sigma_w(\mathcal{A}_k)|})^2. \tag{3.3}$$

(3.3) の等式において，$\sigma_{w_{k_i}}(\mathcal{A}_k) = \{t_{k_i} | (w_{k_i}, t_{k_j}) \in \mathcal{A}_k\}$，$\overline{|\sigma_w(\mathcal{A}_k)|} = \frac{1}{|\mathcal{W}|} \sum_{i=1}^{|\mathcal{W}|} |\sigma_{w_i}(\mathcal{A}_k)|$ であり，それぞれワーカ w_{k_i} に割当てられたタスクの集合と，全ワーカに割当てられたタスク数の平均を表す．

表 1　記号の定義

記号	定義
$\mathcal{W} = \{w_1, w_2, ..., w_n\}$	ワーカ集合
$\mathcal{T} = \{t_1, t_2, ..., t_m\}$	タスク集合
$t_i = (i, d_i)$	タスク (i=タスク番号，d_i=特徴量)
$\mathcal{A}_k = \{(w_{k_m}, t_{k_m})\}$	タスク割当てタプル集合
Th	タスク結果品質の threshold
$d(A_k)$	ワーカ間の割当てタスク数の分散
$ans(w_{k_i}, t_{k_i})$	ワーカ w_{k_i} のタスク t_{k_j} への回答
$\hat{ans}_{k_j}$	タスク t_{k_j} の ground truth
$\delta(ans(w_{k_i}, t_{k_i}) = \hat{ans})$	引数が真のとき 1，偽なら 0 を返す関数
$\sigma_{w_{k_i}}(\mathcal{A}_k)$	ワーカ w_{ki} に割当てられたタスクの集合
$f(d_j)$	タスク t_j への回答を返す AI モデルの関数
$g(f(d_j))$	タスク困難度と相関のある変数を返す関数
$\widehat{\mathcal{W}_{t_j}}$	タスク t_j について割当て先の候補として承認されたワーカ集合
$\mathcal{B}_{w_i}$	ワーカ w_i の $\widehat{\mathcal{W}}$中に登場する回数

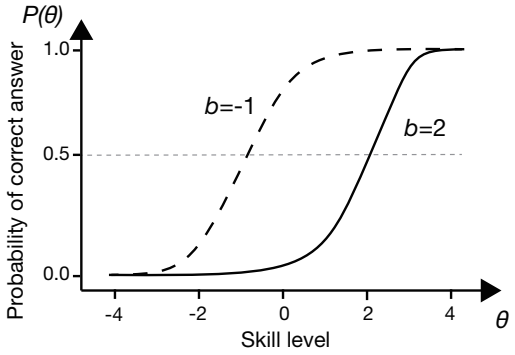

図 4　IRT の項目特性曲線

3.2 項目反応理論

本研究ではタスクの困難度とワーカのスキル分布を求めるために項目反応理論 (IRT) を用いる．IRT によってワーカのスキルを表すパラメータ θ を計算することができる．図 4 は IRT の核となる仮定を表している．IRT では，タスクの困難度 b とワーカのスキル θ が同一の次元上で表現される．つまり，b=θ であるときワーカは 50%の確率でそのタスクに正しい回答を与える．$b > \theta$(または $b < \theta$) であるときはそのワーカがタスクに正しく回答する確率は 50%以下 (以上) である．言い換えれば，困難度が θ 以下のタスクを見つけることができれば，そのワーカはタスクを正しく完了することができるとみなせる．したがって，本研究の提案手法ではワーカの過去のタスク結果の平均正解率のみによってタスクを割当てるときよりも多くのワーカをタスクに取り組ませることができると考えられる．IRT の 1 パラメータロジスティックモデルにおいてワーカ w_i がタスク t_j(IRT では項目と呼ばれる) に正しく回答できる確率は以下の等式で表現される．

Algorithm 1 Proposed framework

Require: $\mathcal{W}$, $\mathcal{T}$, Th, and the AI model $f : \mathcal{D} \to \mathcal{V} \times \mathcal{C}$
Ensure: $\mathcal{A}_k$
1: Apply IRT to $\mathcal{T}' \subseteq \mathcal{T}$ and $\mathcal{W}$ to create a skill test
2: Use the test to estimate the skill θ_i of each worker w_i in $\mathcal{W}$.
3: Obtain $g(f(d_j))$ that correlate with the IRT's difficulty values for $\mathcal{T}'$. $f(d_j)$ returns the task result and its confidence value given by the AI model for t_j
4: Make the task assignment from $\mathcal{T} - \mathcal{T}'$ to $\mathcal{W}$ based on $\{\theta_i\}$ and $\{g(f(d_j))\}$, that satisfies the two conditions.

$$P(ans(w_i, t_j) = a\hat{n}s_j \mid \theta_i, b_j) = \frac{1}{1 + exp(-D(\theta_i - b_j))}, \tag{3.4}$$

この等式において，b_j はタスク t_j の困難度を表すパラメータであり，値が大きいほどそのタスクは難しいということを意味する．D は “scale factor” と呼ばれる定数で，一般的には 1.7 が用いられる．

3.3 フレームワーク

Algorithm1 に提案手法の全体像を示す．Algorithm1 には，入力としてワーカ集合 $\mathcal{W}$ とタスク集合 $\mathcal{T}$，そしてタスク結果品質の threshold Th とタスク内で利用される AI モデルが関数 $f : \mathcal{D} \to \mathcal{V} \times \mathcal{C}$ として与えられる．この関数は，タスク $t_j \in \mathcal{T}$ の特徴 $d_j \in \mathcal{D}$ が入力されたとき，タスクへの回答 $v_j \in \mathcal{V}$ とその回答への確信度 $c_j \in \mathcal{C}$ のタプルを返す．

提案手法のフレームワークでは，はじめに $\mathcal{T}$ のタスクからワーカのスキルを測るテスト用のタスク集合 $\mathcal{T}$’ を作成し，ワーカの回答から $\mathcal{T}'$ の各タスクの困難度 b_j を推定する．次に，そのテストを用いてワーカのスキル θ_i を推定する．また，テストに用いなかった残りのタスク $\mathcal{T} - \mathcal{T}'$ について AI モデルの出力から困難度を推定する．最後に，$\mathcal{T} - \mathcal{T}'$ を推定されたタスク困難度とワーカのスキルに基づき，ワーカに割当てる．

ここで，いくつか注記しておくべき事項がある．第一に，ワーカのスキルを測るためのテストを作成するとき，IRT はテスト用タスクの ground truth，つまり正解が参照可能であることを前提とする．$\mathcal{T}'$ のタスクに ground truth がない場合には，集約された結果を利用することができるものとする． 例えば Dawid-Skene [11] のモデルを使うことでワーカから得られた回答の混同行列からタスクの ground truth を推定することができる．

第二に，$g(f(d_j))$ は任意の関数になりうる．もっとも単純な関数は $f(d_j)$ の返り値の確信度 c_j を出力するというものであり，AI モデルの回答の確信度 c_j と IRT による困難度に相関がある場合は $\mathcal{T}$ に含まれるタスクの困難度推定に利用することができる．

第三に，IRT によるタスクの困難度と相関のある値を返す関数 $g(f(d_j))$ が与えられたとき，各タスク t_j について割当て先の候補として承認されたワーカ (qualified workers) 集合 $\widehat{\mathcal{W}_{t_j}} \subseteq \mathcal{W}$ を求める． 提案手法では $g(f(d_j))$ に対応する推定困難度 b' を回帰的に求める．このとき，$\mathcal{W}_i$ はタスク t_j に Th 以上の確率で正解できるスキルを持つと推定されたワーカの集合であり，次の等式を満たす．

$$\widehat{\mathcal{W}_j} = \{w_i \in \mathcal{W} \mid P(ans(w_i, t_j) = a\hat{n}s_j \mid \theta_i, b_j) \geqq Th\}. \tag{3.5}$$

3.4 Greedy 割当てアルゴリズム

Algorithm 2 Assignment Algorithm

Input: $\mathcal{T} - \mathcal{T}'$, $\mathcal{W}$, $\mathcal{B}_{w_i}$ of each worker w_i, and
$\widehat{\mathcal{W}_{t_j}}$: qualified worker set of each task t_j
Output: $\mathcal{A}_k$
1: $\mathcal{A}_k = \phi$
2: $\mathcal{W}_{assigned} = \phi$
3: **for** $w_i \in \mathcal{W}$ **do**
4: $\quad \mathcal{B}_{w_i}^{init} \leftarrow \{\widehat{\mathcal{W}_{t_j}} \in \mathcal{P}(\mathcal{W}) \,|w_i \in \widehat{\mathcal{W}_{t_j}}\}$
5: $\quad \mathcal{B}_{t_i} \leftarrow \mathcal{B}_{t_i}^{init}$
6: **end for**
7: *sorted_list* $\leftarrow$ *Sort* $\mathrm{t}_j \in \mathcal{T} - \mathcal{T}'$ *according to* $\sum_{w_i \in \widehat{\mathcal{W}_j}} \mathcal{B}_{t_i} / |\widehat{\mathcal{W}_{t_j}}|$
8: **for** $t_j \in$ *sorted_list* **do**
9: $\quad (w_i, t_j) \leftarrow \arg\min_{w_i \in \widehat{\mathcal{W}_{t_j}}} \mathcal{B}_{w_i}$
10: $\quad \mathcal{A}_k \leftarrow \mathcal{A}_k \cup \{(w_i, t_j)\}$
11: $\quad \mathcal{B}_{w_i} \leftarrow \infty$
12: **end for**
13: $\mathcal{W}_{assigned} \leftarrow \mathcal{W}_{assigned} \cup \{w_i\}$
14: **if** $\mathcal{W} = \mathcal{W}_{assigned}$ **then**
15: $\quad$ **for** $w_i \in \mathcal{W}$ **do**
16: $\quad\quad \mathcal{B}_{w_i} \leftarrow B_{w_i}^{init}$
17: $\quad\quad \mathcal{W}_{assigned} \leftarrow \phi$
18: $\quad$ **end for**
19: **end if**
20: return $\mathcal{A}_k$

本研究では上記の目的関数を満たす割当てを見つけるために Greedy なアルゴリズムを用いる．このアルゴリズムでは，実行できるタスクの数が少ない能力の低いワーカに優先的にタスクを割当てる，という戦略をとる．はじめに，(1) 各ワーカ w_i について $\mathcal{T} - \mathcal{T}'$ のタスクの $\widehat{\mathcal{W}}$中に登場する回数B_{w_i} を数える．IRT の性質上，スキルの高いワーカは登場回数が多く，スキルの低いワーカは登場回数が少なくなる．

(2) 次に，各タスクの $\widehat{\mathcal{W}_{t_j}}$ に含まれるワーカの $\mathcal{B}$ の平均値を計算し，平均値が小さいタスクから順にソートする．割当先の少ないワーカが $\widehat{\mathcal{W}_{t_j}}$ に多く出現するタスクを先に割当てることでスキルの低いワーカの優先度を上げることができる．

(3) 各タスクを巡回し，$\widehat{\mathcal{W}_{t_j}}$ 中のワーカから $\mathcal{B}_{w_i}$ の値が最も小さいワーカを選び，タスクを割当てる． 一度タスクを割当てられたワーカは他のワーカにタスクが割当てられるまで残りのタスクの割当て候補から除外する．このアルゴリズムによって，ワーカ間の担当タスク数の分散が最小化されるようなタスク割当ての組み合わせの近似解を求めることができる．

4 実　　験

本研究のリサーチクエスチョンは提案手法がどの程度有効で

あるか，ということである．本論文では現実のクラウドワーカによって完了されたタスクの結果を用いてタスク割当てのシミュレーションを実行した．

4.1 タスクとワーカについて

この実験では，news-aggregator-dataset[2]を利用して 100 件のタスクを設計した ($|\mathcal{T}| = 100$)．それぞれのタスクには，データセットから取り出したニュース記事タイトルのテキストと，そのテキストを AI モデルが Economy，Business，Technology&Science，Health のいずれかのカテゴリへ分類した結果が表示されており，ワーカは AI モデルによる分類結果が正しいかどうかを二択で答える．今回の実験では AI モデルとしてナイーブベイズモデルを利用し，AI モデルによる分類の正解率は 92.8%だった．ワーカからの回答は Amazon Mechanical Turk 上でクラウドソーシング実験を行い収集した．クラウドソーシング実験の結果，ワーカ全員がすべてのタスクに取り組んだ結果の平均正解率は 57.7%であった．

4.2 実験手順

割当てシミュレーションのため，ランダムに $\mathcal{T}'$ のタスクを選択し $\mathcal{T}$ ($|\mathcal{T}| = 100$) を $\mathcal{T}'$ ($|\mathcal{T}'|$=60) と $\mathcal{T} - \mathcal{T}'$ に分割した．その上で，$\mathcal{W}$ に含まれる 98 人のワーカに $\mathcal{T} - \mathcal{T}'$ のタスクを，IRT によるスキル推定に基づいて割当てた． 2 つの目的関数 (Algorithm1，4 行目) を満たすために，割当ての候補となるタスクの数が少ないワーカを優先した greedy な割当てアルゴリズムを実行する．実験では「(1) 少なくとも一つ以上のタスクに取り組むことのできるワーカの人数」，「(2) threshold が変化したときの割当て結果の労働負荷分散と品質」の 2 点をベースライン手法と比較して評価する．

(1) では本研究の提案手法を, Average Accuracy(AA) をベースラインとして，タスクに取り組むことが認められたワーカの人数を比較する．AA は $\mathcal{T}'$ のタスク全体の平均正解率が threshold Th よりも高いワーカにタスク割当てる手法であり，実際のタスク割当てにおいても広く用いられている．(1) を評価する目的は目的はタスクの困難度とワーカのスキルレベルを考慮する提案手法と，ワーカの過去のタスクの平均正解率のみを評価する手法のどちらがより多くのワーカにタスクに取り組む機会を与えられるかを明らかにすることである．

(2) では，各ワーカに割当てられたタスク数，つまり労働負荷の分散とタスク結果の品質が異なる Th の間でどのように推移するかを 2 つのベースラインと比較して評価する．ベースライン手法の一つは各タスクについてランダムなワーカ 1 人を割当てる手法 (random), つまり各ワーカにタスクが割当てられる機会を平等にし，タスクの分担のみを優先する手法である．2 つ目のベースライン手法はタスク T' の正解率が上位 N 人 ($N = 5$) のワーカにのみタスクを割当てる手法 (top) であり，高い結果品質を期待できる代わりにタスクに取り組めるワーカの人数が限定されるような手法である．

実験では提案手法とベースライン手法のタスク割当て結果の品質と，ワーカ間の割当てタスク数の分散を比較する．この比較により, 結果品質と労働負荷分散の片方の指標を優先した割当てに比べて提案手法が 2 つの指標をどの程度両立させることができるか (**RQ1**) を評価することができる．

2：https://archive.ics.uci.edu/ml/datasets/News+Aggregator

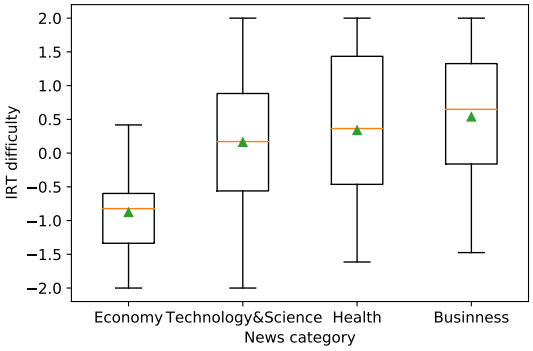

図 5　同じ正解カテゴリを持つタスクの IRT 難易度の分布
(縦軸)IRT による困難度 (横軸) タスクの正解カテゴリ

実験では，0.5 から 0.8 の範囲で 0.01 間隔の各 threshold について 40 回ずつ割当てシミュレーションを繰り返し，イテレーション間の平均を結果として出力した．また，$\mathcal{T}'$ はイテレーションのたびにランダムに選択し直して更新した．

4.3 ワーカ未回答タスクの困難度推定

$\mathcal{T} - \mathcal{T}'$ のタスクはワーカの回答が集まっていないため，IRT によって困難度を求めることができない．したがって，提案手法では AI モデルを利用して $\mathcal{T} - \mathcal{T}'$ のタスクの困難度を推定する．具体的には，IRT によるタスク困難度と相関のある変数を求めるために 2 つのアプローチを試みる．1 つ目のアプローチは (a) AI モデルによって推定された回答の確信度と IRT のタスク困難度との間の相関を利用する方法，2 つ目のアプローチは (b) AI モデルによる分類結果のカテゴリと IRT の難易度との相関を利用する方法である．図 5 の箱ひげ図はこの実験のタスクの各カテゴリごとに異なる困難度の分布を表している．

この実験のタスクでは AI モデルの確信度と IRT による困難度との間に相関が見られなかった． したがって，この実験では．(b) のアプローチを利用して $g(f(d_j))$ を実装する．具体的には AI によるタスクの分類カテゴリと IRT によるタスク困難度との間に十分な相関があると仮定し，$\mathcal{T} - \mathcal{T}'$ のタスクの難易度推定に利用する．特に，$g(f(d_j))$ は AI モデルによってあるカテゴリに分類されたタスクの難易度として各カテゴリの困難度の平均値を返す関数とする．

4.4 実験結果

図 6 は 0.6，0.7，0.8 の各 Th において承認されたワーカ (少なくとも一つ以上のタスクに取り組むことが認められたワーカ) の人数の推移を表している．一般的には，threshold が高くなるほど承認されるワーカの人数は減少するはずである．評価実験の結果では，各 threshold において提案手法によって承認されたワーカの人数がベースライン手法 AA によって承認された

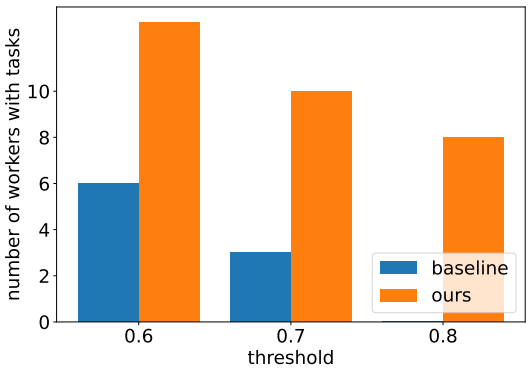

図 6　baseline(AA) と提案手法のタスクに取り組むことができるワーカ人数の threshold 間の推移: 提案手法はより多くのワーカに仕事の機会を与えることができる.

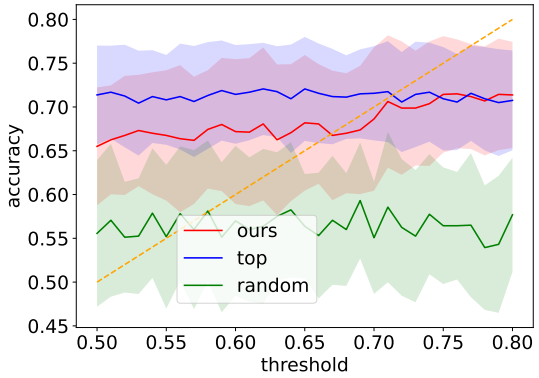

図 7　タスク割当ての結果品質の threshold 間の推移:　提案手法は random より高い品質を維持できる.(縦軸) Accuracy (横軸) threshold

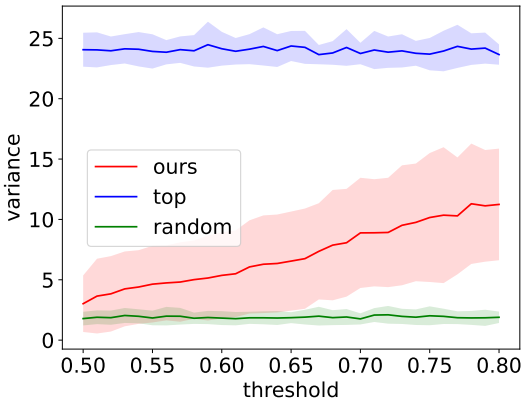

図 8　タスク割当ての労働負荷分散: 提案手法は top よりもワーカ間でタスクを分担できる.(縦軸) Variance (横軸)threshold

ワーカの人数を上回っている．つまり，この結果は本研究の提案手法はより多くのワーカにタスクに取り組む機会を与えることができるという点で効果的であることを示している．図 7 は提案手法とベースライン手法による，ワーカが取り組んだタスク結果の正解率の threshold 間の推移を表している．提案手法

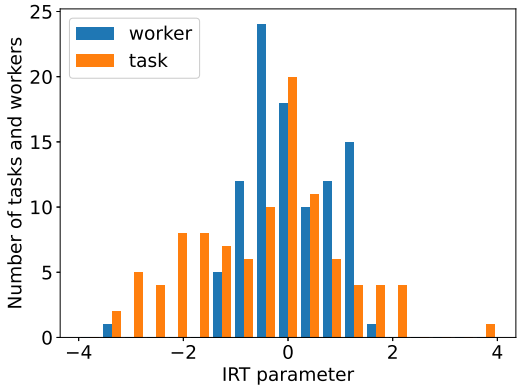

図 9　ワーカとタスクのスキル，困難度ごとの実際の度数分布: 多くのタスクとワーカの分布は重なっている．(青) ワーカ (オレンジ) タスク

はランダムな割当て (random) より高い正解率を維持している．提案手法は上位ワーカのみに割当てる手法 (top) より正解率が低い．top はより高い threshold に対しても品質を満たすことができている．提案手法は threshold が $0.5 \leqq Th \leqq 0.6$ の範囲では正解率を維持できているが，$0.6 \leqq Th \leqq 0.8$ の範囲では正解率が threshold を下回ることが多くなっている．

提案手法と top, random の 2 つのベースライン手法の正解率について，0.5 と 0.8 の 2 つの threshold において統計的な有意差があるかどうかを検定する．この実験では 独立変数を手法，従属変数を正解率とし，Kruskal-Wallis 検定を用いて 3 つの手法間の正解率の差を調べた．その結果，threshold が 0.5 のとき統計的に有意な主効果が認められ ($\chi^2(2) = 59.92, p < 0.05$)，また threshold が 0.8 の場合においても統計的に有意な主効果が認められた ($\chi^2(2) = 54.50, p < 0.05$)．Steel-Dwass 検定を用いた多重比較の結果，0.5 の threshold において提案手法の正解率は random よりも有意に高いことが分かった．さらに，threshold が 0.8 のとき提案手法の正解率は random に比べて有意に高く，提案手法と top の正解率の間には有意な差があるとは言えないことが判明した．以上から，threshold の値の大小によらず提案手法の正解率は random より有意に高く，また値の大きな threshold が与えられたとき提案手法の正解率は top との有意差が認められないほど高くなるということが分かった．

図 8 は提案手法とベースライン手法による，ワーカに割当てられたタスク数の分散の threshold 間の推移を表している．提案手法は上位ワーカのみに割当てる手法 (top) より小さい分散を維持することができている．提案手法はランダムな割当て (random) より高い分散を維持している．

したがって，提案手法は分担のみを優先するランダムな割当てより高い品質を維持しつつ，品質のみを優先する上位ワーカのみへの割当てよりもタスクを分担できることが示されている．

5　考　　察

提案手法によりタスクに取り組むことのできるワーカの人数を増やすことができたが，目的関数を満たすことに関する課題

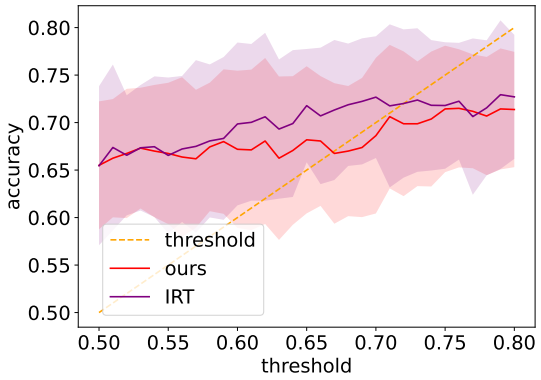

図 10　全てのタスクの難易度を IRT で推定すると提案手法より割当て結果品質を高く維持できる: (縦軸)Accuracy (横軸)threshold

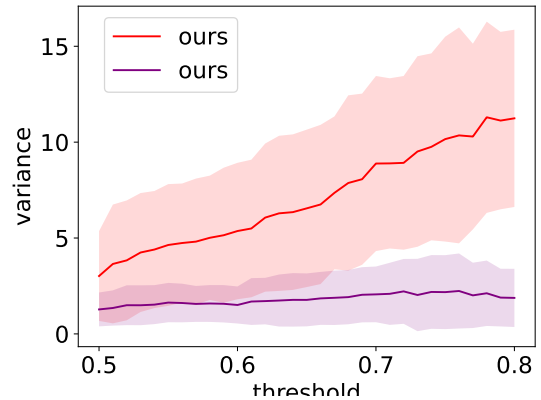

図 11　全てのタスクの難易度を IRT で推定すると労働負荷分散を提案手法よりも小さくできる．(縦軸)variance (横軸)threshold

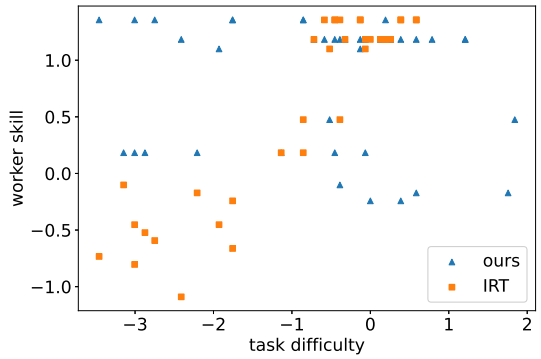

図 12　ワーカに割当てられたタスクの散布図 (横軸) タスクの実際の困難度 (縦軸) タスクを割当てられたワーカの実際のスキル: 提案手法の割当てはタスク困難度とワーカスキルが必ずしも対応しない．

が見つかった．図 9 はワーカのスキルごとの分布と，タスクの困難度ごとの分布を表している．ワーカとタスクの分布する範囲はほとんど重なっている．したがって，ワーカのスキルとタスクの困難度に応じて割当てることができれば高品質のタスク結果を得ることができるはずである．このことを前提として，評価実験の結果について考察する．

評価実験では結果の品質について threshold が一定以上の高さになると結果の品質が threshold 以上に維持できなくなるという結果が得られた．一方，図 10 は提案手法と，全てのタスクの難易度を IRT を用いて推定したうえで割当てる手法 (全 IRT) の結果品質の推移を表している．図 10 の紫色のグラフが全 IRT の結果を表しており，全 IRT に基づくタスク割当ては提案手法より高い threshold においてもタスクの正解率 (accuracy) を維持できている．したがって，正解率の AI モデルによって難易度を推定せずに全てのタスクの IRT 困難度があらかじめ与えられたならば，高い threshold に対しても品質を維持することができるということが示されている．このことから，提案手法のタスク困難度推定方法の精度に問題があると考えられる．また，図 11 では提案手法の割当て結果の分散 (variance) は全 IRT よりもすべての threshold において大きくなっている．つまり，提案手法は全 IRT に比べてワーカ間でタスクを分担できていないということが示されている．

図 12 は $Th = 0.75$ のときの，ワーカに割当てられたタスクの散布図であり，実際にどの程度の困難度のタスクが，どれほどのスキルを持つワーカに割当てられたかということを表している．青い三角の点が提案手法によって割当てられたタスク，オレンジの四角が全 IRT によって割当てられたタスクである．図 12 より，全 IRT では簡単なタスクをスキルの低いワーカに，難しいタスクをスキルの高いワーカに割当てていることができている．しかし，提案手法による割当てではタスクの困難度とワーカのスキルが必ずしも対応していないことが分かる．したがって，提案手法の問題は (1) 実際よりも高い難易度を推定しているタスクがあることと，(2) 実際よりも低い難易度を推定するタスクがあることである．(1) によって実際にはタスクに取り組めるはずのワーカが割当ての候補から除外されるため，タスクに取り組むことのできるワーカの人数が全 IRT に比べて少なくなる． その結果として，提案手法は全 IRT よりもワーカ間の担当タスク数の分散が大きくなっていると考えられる．また， (2) は実際には難しいタスクをスキルの足りないワーカに割当てることになり，割当て結果の品質を下げる原因になっていると考えられる．

上記 (1)，(2) により提案手法の $\mathcal{T} - \mathcal{T}'$ のタスクへの難易度推定方法に問題があることが示されている． 今回の実験の場合は，AI によるタスクの回答 (分類カテゴリ) がタスクの難易度と相関がある，とした仮定が必ずしも正しくなかったことを意味する．本論文の提案手法で結果の品質を維持できなかった範囲の threshold において品質を保つためには，実験結果のタスクの正解率と Th との差，今回の実験結果では 5〜10%程度だけ Th を余分に高く与える必要がある．しかし，threshold を余分に与えるだけ割当て先のワーカの数が制限されるためタスクの分担が難しくなり，提案手法の有効性が下がってしまう．

以上から，目的関数の結果品質の向上，また 割当てタスク数の分散の最小化をより実現するためには，より正確なタスクの難易度推定が可能なモデルが必要である．具体的には，本実験で利用した，AI モデルの出力だけで難易度推定を行う関数

$g(f(d_j))$ ではなく，タスクの他の特徴も利用して推定を行う新たな関数 $g'(d_j)$ が必要であると考えられる．

6 まとめ

本論文では，結果の品質とワーカ間でのタスクの分担の両立を目的として IRT によるワーカのスキル推定に基づいたタスク割当て手法を提案した． 提案手法の評価のため，実際にクラウドソーシングプラットフォームを用いて収集したデータを利用してタスク割当てのシミュレーションを実行した． その結果，提案手法は平均正解率を基にしたタスク割当てよりも多くのワーカにタスクに取り組む機会を与えつつ，労働負荷分散のみを優先する割当てよりも高い品質を得ることができた．しかし，threshold が一定の高さ以上になると結果の品質を保証できなくなることが示され，未だワーカが取り組んでいないタスクについて難易度を推定することの課題が明らかとなった．

今後の展望として (1) より正確なタスクの難易度推定方法の考案，(2) 多次元のスキルへの対応，(3) 割当てアルゴリズムの拡張を計画している．

(1) について，難易度推定の改善のために考えられる方法は，タスクの持つ特徴量から難易度を求める手法 [12] [13] [14] などを導入し IRT による難易度と相関のあるパラメータを求めることである，

また，(2) に関して本論文では IRT は一次元のモデルを用いたが現実に存在する多様なタスクの特徴とワーカのスキルを対応した割当てを実現するためには多次元のスキルへの対応を考慮する必要がある．具体的な方法の一つは多次元項目反応理論 (MIRT) [15] を利用することである。また，特異値分解などの次元削減手法をワーカのタスク回答行列に適用して潜在的なスキルの関連性を分析するという方法も考えられる．ワーカとタスクの多様な関連性見つけることで，一次元のスキル尺度のみを用いる場合に比べてより多くのワーカ間でタスクを分担することが可能になることが期待される．

(3) はより広い観点でワーカ間でタスクを分担するために必要である．本論文ではワーカの労働負荷を単に割当てられたタスクの数としたが，労働時間や報酬の金額などもアルゴリズムに導入することなどが考えられる．

7 謝　辞

本研究の一部は JST CREST JPMJCR16E3 と JSPS 科研費 (JP21H03552) の支援を受けたものである．ここに謝意を示す．

文　献

[1] 森嶋厚行, 喜連川優. クラウドソーシングが不可能を可能にする : 小さな力を集めて大きな力に変える科学と方法. 共立スマートセレクション = Kyoritsu smart selection, No. 32. 共立出版, 2020.

[2] 鹿島久嗣, 小山聡, 馬場雪乃. ヒューマンコンピュテーションとクラウドソーシング = Human computation and crowdsourcing. MLP 機械学習プロフェッショナルシリーズ. 講談社, 2016.

[3] Frank B Baker and Seock-Ho Kim. *Item response theory: Parameter estimation techniques*. CRC Press, 2004.

[4] Hirotaka Hashimoto, Masaki Matsubara, Yuhki Shiraishi, Daisuke Wakatsuki, Jianwei Zhang, and Atsuyuki Morishima. A task assignment method considering inclusiveness and activity degree. In *IEEE BigData 2018 (HMData)*, pp. 3498–3503, 2018.

[5] Yun Huang, Yifeng Huang, Na Xue, and Jeffrey P. Bigham. Leveraging complementary contributions of different workers for efficient crowdsourcing of video captions. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, p. 4617–4626, New York, NY, USA, 2017. Association for Computing Machinery.

[6] Qing Liu, Talel Abdessalem, Huayu Wu, Zihong Yuan, and Stéphane Bressan. Cost minimization and social fairness for spatial crowdsourcing tasks. In *Proc. of DASFAA*, pp. 3–17, 2016.

[7] Fuat Basık, Bugra Gedik, Hakan Ferhatosmanoglu, and Kun-Lung Wu. Fair task allocation in crowdsourced delivery. *IEEE Tran. Serv. Comp.*, 2018.

[8] Ju Fan, Guoliang Li, Beng Chin Ooi, Kian-lee Tan, and Jianhua Feng. Icrowd: An adaptive crowdsourcing framework. In *SIGMOD*, pp. 1015–1030, 2015.

[9] Xiaoni Duan and Keishi Tajima. Improving multiclass classification in crowdsourcing by using hierarchical schemes. In *WWW*, pp. 2694–2700, 2019.

[10] Danula Hettiachchi, Niels van Berkel, Vassilis Kostakos, and Jorge Goncalves. Crowdcog: A cognitive skill based system for heterogeneous task assignment and recommendation in crowdsourcing. *Proc. ACM Hum.-Comput. Interact.*, Vol. 4, No. CSCW2, 2020.

[11] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, Vol. 28, No. 1, pp. 20–28, 1979.

[12] Radu Tudor Ionescu, Bogdan Alexe, Marius Leordeanu, Marius Popescu, Dim P Papadopoulos, and Vittorio Ferrari. How hard can it be? estimating the difficulty of visual search in an image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2157–2166, 2016.

[13] Florian Scheidegger, Roxana Istrate, Giovanni Mariani, Luca Benini, Costas Bekas, and Cristiano Malossi. Efficient image dataset classification difficulty estimation for predicting deep-learning accuracy. *The Visual Computer*, Vol. 37, No. 6, pp. 1593–1610, 2021.

[14] Yu Jiang, Yuling Sun, Jing Yang, Xin Lin, and Liang He. Enabling uneven task difficulty in micro-task crowdsourcing. In *Proceedings of the 2018 ACM Conference on Supporting Groupwork*, GROUP '18, p. 12–21, New York, NY, USA, 2018. Association for Computing Machinery.

[15] Mark D Reckase. Multidimensional item response theory models. In *Multidimensional item response theory*, pp. 79–112. Springer, 2009.