Human+AI Crowd タスク割り当てのための AI ワーカの効率的な評価

神田　智也†　　伊藤　寛祥††　　森嶋　厚行††

† 筑波大学情報学群
†† 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: †s1911475@klis.tsukuba.ac.jp, ††{ito,mori}@slis.tsukuba.ac.jp

あらまし　近年，クラウドソーシングを用いて不特定多数のワーカに AI を開発させることが可能になっている．そのようなクラウドワーカによる AI と人間が共同でタスクに取り組むフレームワークとして Human+AI Crowd が提案されている．従来の Human+AI Crowd に対するタスク割り当て手法においては全ての AI ワーカの提出したタスク結果を毎回評価していた．そのため，タスク割り当ての計算コストが大きくスケーラビリティに難があった．本論文では，要求性能を満たさないと思われる AI ワーカの結果の評価をスキップすることで，タスク割り当ての効率化を図る．我々は 2 つの手法を提案し，その両方が AI ワーカに対するタスク割り当て数をある程度維持しつつ，AI ワーカ評価回数の 90%以上の削減を達成した．これは同じ計算リソースで数十〜数百倍の AI ワーカを利用可能にできることを示している．

キーワード　ヒューマンコンピュテーション，クラウドソーシング，Human-in-the-loop

1　はじめに

近年，Amazon Mechanical Turk[1]等のクラウドソーシングプラットフォームを利用してクラウドワーカに大量のタスクを依頼することは一般的である．クラウドワーカには直接タスクを取り組ませるだけでなく，AI の開発，例えば分類タスクを行う機械学習モデルの開発を依頼することも可能である．Kaggle[2] のようなサイトを用いることで，大量のタスクを解く AI の開発を依頼し，AI によって解かれたタスク結果を得ることができる．

このように，クラウドワーカが開発する AI を容易に利用できる現在，人間ワーカとクラウドによって作られた “AI ワーカ”をどのように適切にタスクに割り当てるのか，そのためのタスク割り当てアルゴリズムの再考が必要であると指摘されている [1].

そこで，我々は多数の人間のクラウドワーカーと多数のクラウドソーシングで得られる AI が共にタスクに取り組む “Human+AI Crowd”に対するタスク割り当てアルゴリズムを提案している [6]．Human+AI Crowd は人間が直接タスクに取り組む “人間ワーカ”と，クラウドワーカが開発したタスクを遂行する AI である “AI ワーカ” [3] の集団である（図 1）．論文 [6] で我々が提案した手法は，AI のモデルに関する情報が分からない “ブラックボックス”AI ワーカを想定している．我々の想定する状況は最初はタスクの結果が 1 つも無い状態からランダムに人間ワーカにタスクを割り当てていくものである．その後，

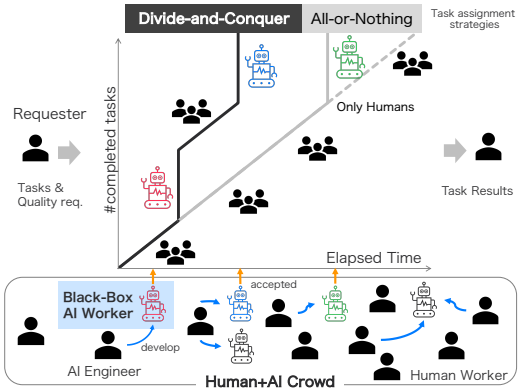

図 1　Human+AI Crowd におけるタスク割り当ての図（[6] より引用）．マイクロタスクが人間のクラウドワーカだけでなくクラウドワーカーによって作成されたブラックボックスな AI ワーカにも割り当てられる．AI ワーカは人間ワーカのタスク結果を利用してモデルを訓練し，訓練されたモデルが他のタスクに対する結果を出力する．タスク割り当てシステムは AI ワーカの出力した結果が十分な精度がある場合，AI ワーカの出力した結果を採用する．タスク割り当てシステムはより早く要求品質を満たした AI ワーカの結果を採用する．

人間ワーカの提出したタスク結果がある程度集まるごとに，それをもとに AI ワーカを学習させる．そして，要求を満たすタスク結果を出力した AI ワーカを採用するという状況を想定している．論文 [6] ではこの状況で，タスク全体の品質を維持しつつ AI ワーカに割り当てるタスク数を最大化する問題を HACTAP（Human+AI Crowd Task Assignment Problem）として定義した．

さらに論文 [6] では分類タスクにおいてタスククラスタという

1：https://www.mturk.com/
2：https://www.kaggle.com

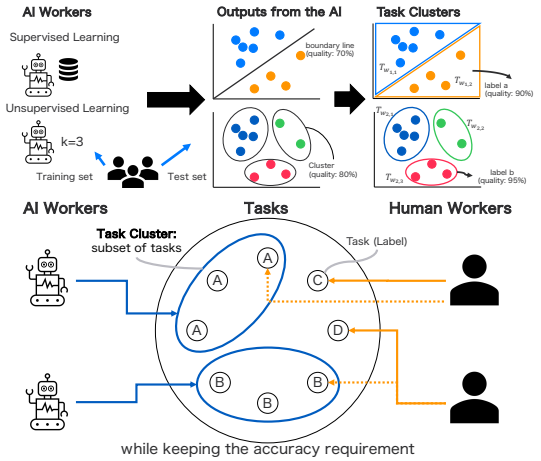

図 2 [6] で我々が提案したタスククラスタの概要．各 AI ワーカの出力は，その AI ワーカが同じラベルであると出力したタスクの集合であるタスククラスタという単位で扱われる（上部）．最初は人間ワーカにランダムに割り当てられるが，タスククラスタが統計的検定によって要求精度を満たすと判断された場合，タスククラスタに含まれるタスクが AI ワーカに割り当てられる（下部）

概念を導入することでこの問題に取り組んだ．タスククラスタは AI ワーカーが同じラベルであると予測したタスクの集団である．従来の手法では AI ワーカの出力する結果の全体に対して採用するか否かを判断していた．しかし，タスククラスタの導入によって AI ワーカを部分的に採用するということが可能になった．タスククラスタを利用したタスク割り当てアルゴリズムとして我々は CTA（Clusterwise Test-based Assignment）と GTA（Global Test-based Assignment）を提案した．

ここで問題となるのが AI ワーカの評価である．人間ワーカはタスク結果の精度が低ければ低品質なワーカーもしくはスパムワーカであると判断できる．しかし，AI ワーカは単に精度が低いだけではスパム等と判断することはできない．なぜならその AI ワーカがまだ学習の途中である可能性や，一部の分類ラベルのみに対して高精度を出すことができる可能性があるからである．そのため，既存手法においては全ての AI ワーカのタスク結果を毎回評価して採否を判断していた．

図 2 はタスククラスタを利用した手法の概要を示している．それぞれの AI ワーカのタスク結果は同じラベルに分類された要素ごとにタスククラスタとして扱われている（図 2 上部）．HACTAP において，人間ワーカもしくは AI ワーカに取り組ませるマイクロタスクの集合を $\mathcal{T}$ とする（図 2 下部）．HACTAP ではまず，ランダムに選ばれたタスクが人間ワーカに割り当てられる．人間ワーカに割り当てたタスクの結果がある程度集まったら，アルゴリズム 1 に示す割り当てアルゴリズムを実行して AI ワーカの訓練と AI ワーカに対するタスク割り当てを行う．（記号類は表 1 に示す）．アルゴリズムの 2 行目で各 AI ワーカ w_i の各タスククラスタ $\mathcal{T}_{w_i,j}$ $(j = 1, 2, \ldots)$ が精度を満たすか否か判断する．我々の提案したタスク割り当て手法の 1 つ，CTA においては各 AI ワーカ w_i の各タスククラスタがそれぞれ $\mathcal{T}_{w_i,j}$ に対してそれぞれ要求精度を満たしているか

Algorithm 1 Evaluation Step of Task Clusters in CTA/GTA

Input: A set W of AI workers w_i, a set $\mathcal{C}$ of task clusters $T_{w_i,j}$s of all AI workers, accuracy requirement α, a sequence $\mathcal{A}$ of the task results for a subset of $\mathcal{T}$ given by crowd human workers so far.
Output: A set $\mathcal{T}' \subseteq \mathcal{T}$ of tasks that AI workers will complete.
1: **for all** $\mathcal{T}_{w_i,j} \in \mathcal{C}$ **do**
2: **if** $PassTheStatisticalTest@HACTAP(\mathcal{T}_{w_i,j}, \mathcal{A}, \alpha)$ **then**
3: Add tasks in $\mathcal{T}_{w_i,j}$ to T'
4: **end if**
5: **end for**
6: **return** $\mathcal{T}'$

統計的検定を行う．我々のもう一つの手法，GTA においては，$\mathcal{T}_{w_i,j}$ を採用した場合のタスク全体の精度が要求精度を満たすかモンテカルロシミュレーションを行う．タスククラスタが精度を満たすと判断されたら，そのタスククラスタ内のタスクが AI ワーカに割り当てられ（図 2 下部），その結果が出力される（アルゴリズム 1 の 3 行目）．

しかし，彼らの手法では大量のタスククラスタを評価する必要があるため，スケーラビリティが低い．彼らの手法においては一定数の人間ワーカからのタスク結果を受け取るたびに全ての AI ワーカの全てのタスククラスタを評価している．そのため，その計算量は $O(|\mathcal{T}| \cdot |\overline{\mathcal{C}}|)$ になる．ここで $|\mathcal{T}|$ はタスク数，$|\overline{\mathcal{C}}|$ は毎回評価するタスククラスタ数の平均を表す．$|\overline{\mathcal{C}}|$ は $|\mathcal{W}|$ すなわち AI ワーカの数に比例する．その上，GTA はモンテカルロシミュレーションを利用する [6] ため，その計算コストは大きい．

本論文 [3] は AI ワーカを効率的に評価する手法を提案するものである．AI ワーカは HACTAP の初期の段階においてはモデルの訓練が十分でないために要求精度を満たせない場合が考えられる．しかし，それだけでは必ずしもスパム AI ワーカとは言い切れないため，一度要求精度を満たせないだけでその AI ワーカの評価を打ち切ることは不適切である．そこで，我々は AI ワーカが要求精度を満たせない場合はその AI ワーカの評価をスキップし，後に評価を再開することを考える．しかし，評価を再開するタイミングを決定するにあたって，AI ワーカがブラックボックスであるということが問題となる．そのため我々は AI ワーカに対する入出力のみに基づいて AI ワーカをスキップする期間を決定する必要がある．我々はこの問題に対して 2 つの手法を提案する．1 つ目に AI ワーカが要求精度を満たせない場合に，次回評価までの期間つまり評価スキップ回数を倍にする手法，double time-interval strategy（DTI）を提案する．2 つ目に提案する手法は AI ワーカの出力するタスク結果からその AI ワーカの学習曲線を推定し，その学習曲線を利用して適切な次回評価のタイミングを決定する learning-curve estimation strategy（LCE）である．

我々は 2 つの手法とベースライン手法を比較する実験を行い，我々の手法の両方が AI ワーカに割り当てるタスク数をほとんど減らすことなく評価回数を 90%以上削減可能であることを示した．

本論文の貢献（1） 2 つの提案手法は AI ワーカの評価回数を

3：本論文は [5] の内容を拡張したものである．

表 1　記号類の定義

記号	意味
$\mathcal{W}$	AI ワーカの集合
$\mathcal{T}$	タスク集合
$\mathcal{A}$	人間ワーカから得られた $\mathcal{T}$ の回答の集合
$\mathcal{T}_{w_i,j} \subseteq \mathcal{T}$	AI ワーカ w_i の j 個目のタスククラスタ
$\mathcal{C} \subseteq 2^{\mathcal{T}}$	タスククラスタの集合
$\mathcal{S}_{w_i} \in \{0,1\}$	各 AI ワーカの前回評価の配列
$R_{w_i} \in \mathbb{Z}^+$	各 AI ワーカが評価で否決された回数の配列
$N_{w_i} \in \mathbb{Z}^+$	各 AI ワーカの次回評価タイミングの配列
c_{iter}	学習のイテレーションカウンター
$\mathcal{LC}_{w_i}$	各 AI ワーカの推定された学習曲線の配列
$\alpha \in [0,1]$	HACTAP における有意水準
$\alpha_{AI} \in [0,1]$	AI 評価における有意水準
$q_{AI} \in [0,1]$	AI ワーカ評価における AI に対する要求性能

90%以上削減することで Human+AI Crowd に対するタスク割り当てのスケーラビリティを向上させられることを示した．(2) 一見単純な DTI（Double Time-Interval）が AI ワーカへのタスク割り当て数を維持するという点において，AI ワーカの学習曲線を推定する高度な手法である LCE（Learning-Curve Estimation）より良好な性能を示すことが実験によって明らかになった．

2 関 連 研 究

2.1 Human+AI Crowd Task Assignment Algorithm

我々は [6] でマイクロタスクが人間とブラックボックスな AI ワーカに割り当てられる Human+AI Crowd の概念を提案し，可能な限り多くのタスクを AI ワーカに割り当てる手法を考案した．従来の手法では AI ワーカがタスク集合全体に対して要求性能を満たすまで AI ワーカが採用されない，いわば All-or-Nothing 的な AI ワーカの評価を行っていた．我々の提案したタスク割り当て手法は All-or-Nothing 的な AI ワーカの評価を排するためにタスククラスタの概念を導入した．タスククラスタは各 AI ワーカが同じラベルに分類したタスクの集合であり，タスク全体の集合の部分集合である．本論文では彼らの手法を改善するために要求性能を満たさないと推測される AI ワーカの評価をスキップすることを提案する．

我々は [6] でタスククラスタを利用した Human+AI Crowd に対するタスク割り当て手法として CTA と GTA を提案した．**CTA** CTA はタスククラスタをそれぞれ統計的検定にかけタスクの要求品質を満たすタスククラスタに含まれるタスクを AI ワーカに割り当てる手法である．しかしながら，CTA はタスククラスタごとに独立に統計的検定を行うために，採用されるタスククラスタの数が多くなるほど第 1 種過誤の危険が高まることは我々の論文 [6] 内でも指摘した通りである．最終的に採用されるタスククラスタの個数は分からないため，有意水準に補正をかけることも困難である．もしくは，オンライン統計的検定の手法の開発が必要である．

GTA 我々は CTA の問題点を受け，統計的検定の回数を減らすことで第 1 種過誤の危険を減らした手法である GTA を提案した．GTA は採用候補のタスククラスタをすでに採用されているタスククラスタの集合に加え，その集合をモンテカルロシミュレーションによって評価する．これによって採用されたタスククラスタの集合に対する統計的検定の回数を減らし，第 1 種過誤の危険を低減した．

しかし毎回各採用候補タスククラスタに対してモンテカルロシミュレーションを行うために，GTA はタスク割り当てにかかる計算コストが大きい．本論文では，GTA の計算コストを改善する手法を提案する．

2.2 学習曲線の推定

機械学習モデルの学習曲線の推定に関する研究は数多くある．例えば，機械翻訳モデルに対する実用的な学習曲線推定手法 [7] や，偏ったデータに対する限られたサンプルで学習したロジスティック回帰モデルの学習曲線の推定に関する研究 [9] などがある．しかしながら，異なるモデルの学習曲線の推定に関してそれらを広くカバーする包括的な理論，手法を提案することは難しいとされている [10]．

本論文では [8] で複数の論文で良好な性能を示したとされている単純な学習曲線モデルである $C(x) = ax^{-b} + c$ を採用する．学習曲線の推定時より以前の各 AI ワーカのテストデータ（HACTAP においては，人間ワーカのタスク結果）に対する精度を用いてこのモデルに対して曲線近似を行うことで学習曲線を推定する．

2.3 プラットフォームによる AI ワーカの評価

より高品質な AI モデルを得るためには，クラウドワーカに要求精度を満たさないタスク結果を提出しないことによるインセンティブを与える [4] という方法が考えられる．これを利用してクラウドワーカに高品質な AI ワーカのみを提出させるように促すことで，要求精度を満たさないと考えられる AI ワーカを評価する回数を減らし，タスククラスタの適合率 [4] を向上させることも考えられる．しかし，本論文では，高品質な AI モデルを得る方法として，高品質 AI モデルを得るためにクラウドワーカ側に働きかけるのではなく，タスク割り当てプラットフォーム側が AI ワーカの評価をするという状況を考える．

3 提 案 手 法

3.1 Task Cluster Filters

本論文で我々は AI ワーカの提出するタスククラスタの数を減らすフィルタを提案する．我々は 2 種のフィルタ, DTI（Double Time-Interval Filter）と LCE（Learning-Curve Estimation Filter）を提案する．本節では，各種定義とフィルタの詳細について説明する．

4：タスククラスタの適合率は 3.1.1 節で定義する．

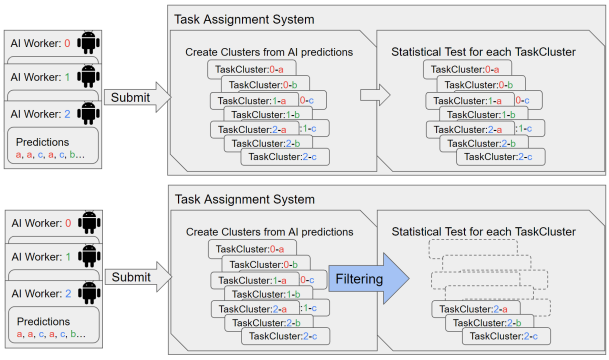

図 3　タスククラスタフィルタなしの HACTAP （上部） とタスククラスタフィルタのある HACTAP （下部）．（[5] から引用）

3.1.1 定　　義

タスククラスタの適合率まず，タスククラスタの適合率を定義する．ここでの適合率とは，GTA においてモンテカルロシミュレーションにおいて評価されるタスククラスタのうち，実際に採用されるタスククラスタの割合を示す．アルゴリズム 1 において，タスククラスタの適合率 $precision(\mathcal{C})$ は次のように定義される．

$$\frac{|\{\mathcal{T}_{w_i,j} \in \mathcal{C} | PassTheStatisticalTest@HACTAP(\mathcal{T}_{w_i,j}, \mathcal{A}, \alpha)\}|}{|\mathcal{C}|}.$$

我々の目的はタスククラスタの集合 $\mathcal{C}$ からタスククラスタを抽出して適合率が高いタスククラスタの集合 $\mathcal{C}'$ を出力するタスククラスタフィルタを開発することである．フィルタが高い適合率を達成することで，$PassTheStatisticalTest@HACTAP(\mathcal{T}_{w_i,j}, \mathcal{A}, \alpha)$ の計算コストを削減することができる．しかし，HACTAP は AI ワーカに割り当てるタスク数を最大化する問題設定であるから，AI ワーカに割り当てるタスク数を維持するために，フィルタは再現性を損なってはいけない．

タスククラスタフィルタ次にタスククラスタフィルタ（TCF: Task Cluster Filter）を定義する．タスククラスタフィルタ f はタスククラスタの集合 $\mathcal{C}$ を入力に受け取り，$\mathcal{C}$ の部分集合である $\mathcal{C}'$ を出力する．タスククラスタフィルタはアルゴリズム 1 の 1 行目に条件節として挿入される．すなわち，要求性能 q_{AI} でタスククラスタをフィルタリングするタスククラスタフィルタ f を導入すると 1 行目が “**for all** $\mathcal{T}_{w_i,j} \in f(\mathcal{C}, q_{AI})$ **do**” となる．

繰り返しになるが，タスククラスタフィルタ $\mathcal{C}' = f(\mathcal{C})$ の目的はタスククラスタの評価の回数を減らすことであり，そのためにもとのタスククラスタ集合 $\mathcal{C}$ から HACTAP の統計的検定を通過しないと思われるタスククラスタを取り除くことである．

3.1.2 設 計 方 針

本論文でのタスククラスタフィルタの設計には 2 つの方針がある．

1 つ目に，我々は各 AI ワーカはブラックボックスであると仮定していることが挙げられる．つまり，AI ワーカが実際にどのように実装されているか我々は知ることができず，入出力だけ分かるということである．ブラックボックスであるとは，具体的には，CNN を利用した AI ワーカや教師なし学習モデルと利用した AI ワーカ，ルールベースの AI ワーカも考えられるということである．それゆえ，我々は各 AI ワーカの AI モデルの内部的な状態に関する情報なしにタスククラスタフィルタを設計する必要がある．

2 つ目に，タスククラスタフィルタは AI ワーカの評価をある間隔でスキップするようにする．すなわち，一度フィルタがその AI ワーカを評価しないと決めたら，しばらくの間その AI ワーカは再評価されないということである．ここで重要となるのが，いつ AI ワーカの評価を再開するべきかということである．

我々は AI ワーカの評価をスキップする戦略について異なる 2 つのフィルタを提案する．

3.2 Filter 1: Double Time-Interval Filter

このフィルタは AI の要求性能 q_{AI} での統計的検定を通過する $\mathcal{T}_{w_i,j}$ （AI ワーカ w_i によって生成されたタスククラスタ）が 1 つも無い場合，全ての $\mathcal{T}_{w_i,j}$ の次回評価をスキップするというものである．このフィルタでは，AI ワーカの評価スキップ間隔を 1, 2, 4, 8, . . . と倍々にしていく．

評価の間隔を倍々にするのは，AI ワーカの性能の改善率は徐々に鈍化していくと考えられるためである．

アルゴリズム 2 に DTI の具体的な処理を示す．入力としてタスククラスタの集合である $\mathcal{C}$ と AI ワーカに対する要求精度として q_{AI} を受け取る．また，DTI 内の統計的検定の有意水準として α_{AI} を受け取る．加えて，DTI は各 AI ワーカ w_i に関するグローバル変数として，次回評価タイミング N_{w_i} とフィルタを通過しなかった回数 R_{w_i} を保持する．

このフィルタ（アルゴリズム 2）は AI ワーカ w_i ごとにフィルタをかける（2 15 行目）．もし該当 AI ワーカがまだスキップ中なら，その AI ワーカから生成されたタスククラスタは全てフィルタを通過しない（3 5 行目）．そうでないならば，該当 AI ワーカから生成された各タスククラスタに対して（6 行目）統計的検定を行い（7 行目）[5]，それに通過したタスククラスタを生成した AI ワーカから生成された全ての AI ワーカはフィルタ出力に追加される（8 行目）．そして次回評価のタイミングは次の次のイテレーションに設定される（つまり，評価をスキップしない）．該当 AI ワーカから生成されたタスククラスタが 1 つも統計的検定に通過しなかった場合，その AI ワーカから生成された全てのタスククラスタはフィルタを通過せず，フィルタを通過しなかった回数のカウンタ R_{w_i} がインクリメントされ（13 行目），次回評価のタイミングは今までのスキップ回数の倍になる（14 行目）．

3.3 Filter 2: Learning-Curve Estimation Filter

AI ワーカの評価とスキップ回数を同時に決定する手法として，我々は学習曲線の推定による AI ワーカの評価を提案する 3．AI ワーカのこれまでの精度から学習曲線を推定し，要求精

5：実験の節で述べているように，本論文で行った実験ではこの統計的検定として二項検定を利用した．

Algorithm 2 Double Time-Interval Filter (DTI)

Input: (Common parameters) A set $\mathcal{C}$ of task clusters and the accuracy requirement for AI worker q_{AI} as common parameters; (Parameters specific to this filter) the significance level α_{AI}; (Global variables) the iteration count N_{w_i} for which AI worker w_i will be evaluated next time and the number R_{w_i} of rejections the AI workers encountered so far for each w_i.
Output: Set $\mathcal{C}_p$ of task clusters which passed the filter (i.e., a subset of $\mathcal{C}$).

1: $\mathcal{C}_p \leftarrow \phi$
2: **for all** $w_i \in \mathcal{W}$ **do**
3: **if** $N_{w_i} > c_{iter}$ **then**
4: continue
5: **end if**
6: **for all** $\mathcal{T}_{w_i,j} \in \mathcal{C}$ **do**
7: **if** $statistical\_test(\mathcal{T}_{w_i,j}, q_{AI}, \alpha_{AI})$ **then**
8: Add all $\mathcal{T}_{w_i,j}$s for w_i to $\mathcal{C}_p$
9: $N_{w_i} \leftarrow n_{iter} + 1$
10: break
11: **end if**
12: **end for**
13: $R_{w_i} \leftarrow R_{w_i} + 1$
14: $N_{w_i} \leftarrow n_{iter} + 2^{R_{w_i}}$
15: **end for**
16: **return** $\mathcal{C}_p$

度を満たすと考えられるタイミングまで評価をスキップするという手法である（図 4）.

学習曲線の推定には特定のモデルに曲線近似するという手法を用いた．回帰のためのモデルとしては $C(x) = ax^{-b} + c$ $(a, b, c > 0)$ を用いる．学習曲線の推定に関して様々な AI モデルを包括的に扱う手法は提案されていない [10] ため，Black-box な AI ワーカに対して特定のモデルにフィッティングさせるということは適切とはいえない．しかし，今回は実際のクラスタ採用のための判定ではなく厳密な推定は求められないため，この手法を採用した．

また，今回は $C(x) = ax^{-b} + c$ を用いることですべての AI ワーカの学習曲線が Well-behaved な学習曲線であること，すなわち学習の進行における精度の低下が起こらないことを暗に仮定している点は注意するべき点である．実際のモデルは学習の過程で精度が一時的に低下する場合も考えられるが，今回は厳密な推定は求めていないため考慮しない．

学習曲線の推定は計算コストが多少あるため毎回すべての AI ワーカに対して行うことは望ましくない．しかし，要求精度を満たすと予測されるまでの間にも AI ワーカは学習による精度向上を果たしていると考えられるため，推定学習曲線を更新しないと，精度向上した AI ワーカのより早い採用の機会を失う可能性がある．そのため，学習曲線の推定の更新は，現在から要求精度を満たすとされるタイミングまでの半分の位置で行うこととする．

LCE の処理の詳細をアルゴリズム 3 に示す．このフィルタも各 AI ワーカ w_i ごとにフィルタリングする（2 11 行目）．もし該当 AI ワーカがまだスキップ中なら，その AI ワーカから生成されたタスククラスタは全てフィルタを通過しない（3 5 行目）.

そうでないなら，学習曲線の推定を行う（6 行目）[6]．推定した学習曲線を利用していつ該当 AI ワーカの評価を再開するべきか決定する（7 行目）．もし AI ワーカが AI ワーカに対する

6：曲線近似に失敗する場合もあるが，その場合は DTI の統計的検定を使う．その場合，スキップ回数を倍々にすることはしない．詳細はここでは省略する．

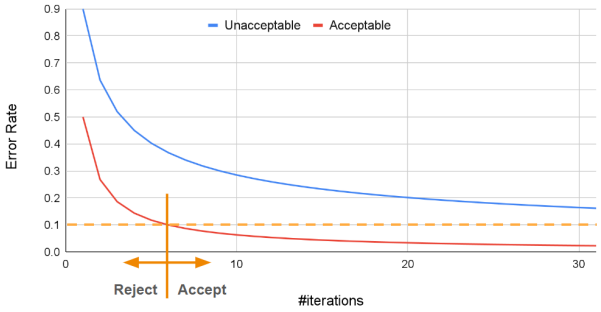

図 4　要求エラーレートが 0.1 の場合に採用されうる AI ワーカの学習曲線（Acceptable）とそうでない学習曲線（Unacceptable）.（[5] から引用）

Algorithm 3 Learning-Curve Estimation Filter (LCE)

Input: (Common parameters) A set $\mathcal{C}$ of task clusters, and the accuracy requirement for AI worker q_{AI}; (Global variables) the iteration count N_{w_i} for which AI worker w_i will evaluated next time and each w_i.
Output: Set $\mathcal{C}_p$ of task clusters which passed the filter (i.e., a subset of $\mathcal{C}$).

1: $\mathcal{C}_p \leftarrow \phi$
2: **for all** $w_i \in \mathcal{W}$ **do**
3: **if** $N_{w_i} > c_{iter}$ **then**
4: continue
5: **end if**
6: $\mathcal{LC}_{w_i} \leftarrow EstimateLearningCurve(w_i)$
7: $N_{w_i} \leftarrow When2Restart(\mathcal{LC}_{w_i}, q_{AI})$
8: **if** $c_{iter} \geqq N_{w_i}$ **then**
9: Add all $\mathcal{T}_{w_i,j}$s for w_i to $\mathcal{C}_p$
10: **end if**
11: **end for**
12: **return** $\mathcal{C}_p$

Algorithm 4 When2Restart

Input: Information of learning curve $\mathcal{LC}_{w_i}$, a number of iterations of learning c_{iter}, the accuracy requirement for AI worker q_{AI}.
Output: A number of next iteration updating learning curve of AI worker.

1: **if** $\mathcal{LC}_{w_i}(c_{iter})$ satisfy q_{AI} **then**
2: **return** $c_{iter} + 1$
3: **else**
4: $p \leftarrow whenItReachesTheRequiredQuality(\mathcal{LC}_{w_i}, q_{AI})$
5: **return** $(p - c_{iter})/2 + c_{iter}$
6: **end if**

要求性能 q_{AI} を満たしているなら， $c_{iter} \geqq N_{w_i}$ が真となり（8 行目），w_i から生成された全てのタスククラスタが出力に加えられる（9 行目）.

評価再開タイミングを決定する関数 When2Restart （アルゴリズム 4）は以下のようなものである．AI ワーカの推定学習曲線がすでに要求性能を満たせているなら（1 行目），次回もまたその AI ワーカを評価するようにする（2 行目）．そうでない場合，評価を再開するタイミングを AI ワーカが要求性能を満たすと学習曲線から推測される位置（4 行目）までの半分の位置に設定する（5 行目）.

学習曲線の推定はパラメータの数だけ，今回は 3 つの精度のデータがあれば可能である．しかし，学習初期の精度は悪いため，ある程度学習が進む前に推定してしまうと，かなりエラーレートの高い学習曲線を推定してしまい，かなり長い間，すべてのタスクが割り当て終了するまで要求水準を満たさない学習曲線を推定する恐れがある．そのため，最初の数回は学習曲線の推定を行わず，DTI の節で述べたタスククラスタに対する統計的検定による判定を代わりに行うことにした.

表 2　実験の条件

パラメータ	条件
フィルタリング手法	NONE（ベースライン）, DTI, LCE
q	0.8, 0.85, 0.9, 0.95
q_{AI}	0.8, 0.9

4　実　　験

DTI と LCE 及びベースライン手法としてのタスククラスタフィルタを用いない手法を比較する実験を行った．本実験でのフィルタの望ましい挙動は不必要なタスククラスタを取り除き（つまり，適合率の向上），AI ワーカに割り当てるタスク数を維持することである．

実験は我々が [6] で利用したコード [7]に手を加えたもの [8]を利用している．我々の提案した GTA のプログラムに我々の提案する AI ワーカの事前評価フィルタ：タスククラスタフィルタを導入した GTALimit を実装した．データセットに [6] 内でベンチマークデータセットとして用いられている Kuzushiji-MNIST [2] の 10 クラス分類を用いた．加えて，Fashion-MNIST [11] を用いた実験を行った．

4.1　HACTAP の設定

HACTAP では「どれだけ人間ワーカのタスク結果が集まるごとに AI ワーカを訓練・推測させ，それを評価して AI ワーカにタスクを割り当てるか」の間隔と，AI ワーカにタスクを割り当てる際の統計的検定，StatisticalTest@HACTAP（今回は GTA ベースであるからモンテカルロシミュレーション）の有意水準を決める必要がある．本論文では [6] の実験で用いられた設定と同じ設定を用いる．AI ワーカは人間ワーカのタスク結果が 200 個集まるごとに訓練，評価される．統計的検定の有意水準は 0.05 に設定した．

AI ワーカは異なる 15 種のモデルを用いた．AI ワーカの詳細は Appendix に記した．

4.2　Double Time-Interval Filter の設定

DTI の AI ワーカ評価のための統計的検定として，二項検定を有意水準 $\alpha_{AI} = 0.05$ で用いた．

4.3　Learning-Curve Estimation Filter の設定

学習曲線の推定には scipy[9]の $curve\_fit$ 関数を用いた．本論文では推定されたパラメータの最適値を用いる．LCE ではアルゴリズム 4 の 1 行目で推定学習曲線が要求エラーレート：$error\_rate = 1 - q_{AI}$ を下回っている場合にその AI ワーカを通過させる．

推定に十分な精度のデータを得るために，HACTAP 開始から $n\_skip\_init = 10q_{AI} + 3$ の間は推定を行わず，q_{AI} の要求精度，α_{AI} の有意水準でで二項検定を行う．

7：https://github.com/crowd4u/HACTAP-Framework
8：https://github.com/crowd4u/HACTAP-Framework/releases/tag/HMData2022v1.3
9：https://scipy.org/

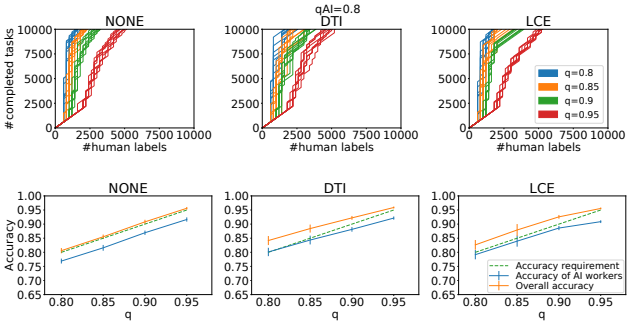

図 5　（Kuzushiji-MNIST, $q_{AI} = 0.8$）上部：AI ワーカへのタスク割り当ての過程．下部：タスク全体における精度．

4.4　条　　件

本実験では次の 3 つの条件間の比較を行った：　“NONE”（フィルタなし），“DTI”（Double Time-Interval Filter），“LCE”（Learning-Curve Estimation Filter）．各条件下で HACTAP におけるタスク全体の要求品質である q と，タスククラスタフィルタのフィルタの閾値である q_{AI} について複数の条件で各 10 回の実験を行い，比較した．条件の詳細は表 2 に示した．

5　結　　果

5.1　タスク割り当て過程とタスク精度

図 5，図 6，図 7，図 8 にそれぞれ，人間ワーカと AI ワーカに対するタスク割り当て過程と，全タスク完了後のタスク全体の精度を示す．上のグラフは人間ワーカによって行われたタスク数と Human+AI Crowd 全体で完了したタスク数を示している．ある時点での AI ワーカの行ったタスク数は縦軸の値から横軸の値を減じたものとなる．$q_{AI} = 0.9$ における LCE の，特に $q = 0.95$ において，NONE と比較して AI ワーカに割り当てられるタスク数が大きく減少している．q_{AI} パラメータによるフィルタの厳しさによって AI ワーカへのタスク割り当て数とフィルタで弾く AI ワーカの数の間にトレードオフ関係が発生している．詳細は後ほど 5.4 節で述べる．

下のグラフはタスク全体の精度と AI ワーカの精度を示している．（エラーバーは標準偏差を示す．）HACTAP はタスク全体の精度を保証するものである [6] から，フィルタを通してもタスク全体の精度が要求精度を下回っていないことが示されている．本論文で提案したフィルタの影響により，全体的に AI ワーカの精度の向上が見られる．

5.2　タスククラスタ評価回数

図 9 は各条件における HACTAP がタスククラスタを評価した回数（すなわち，フィルタを通過したタスククラスタの数）を示している．当然，$q_{AI} = 0.9$ の方が $q_{AI} = 0.8$ の条件よりもタスククラスタの評価回数は少ない．しかし，$q_{AI} = 0.8$ であっても各条件において DTI と LCE は NONE よりも大幅に評価回数を減らすことに成功している．

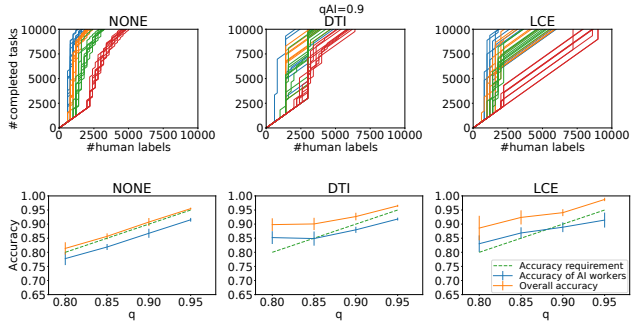

図 6　（Kuzushiji-MNIST, $q_{AI} = 0.9$）上部：AI ワーカへのタスク割り当ての過程．下部：タスク全体における精度．

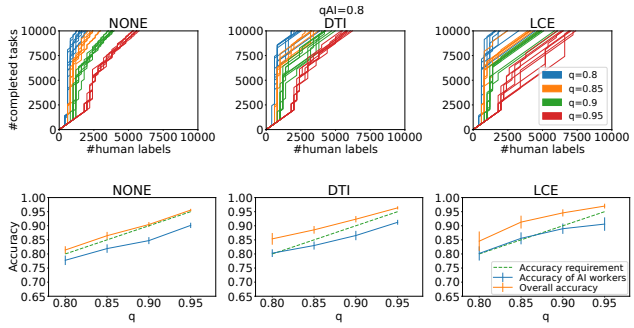

図 7　（Fashion-MNIST, $q_{AI} = 0.8$）上部：AI ワーカへのタスク割り当ての過程．下部：タスク全体における精度．

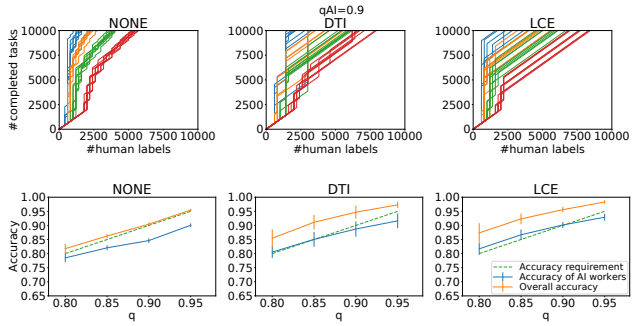

図 8　（Fashion-MNIST, $q_{AI} = 0.9$）上部：AI ワーカへのタスク割り当ての過程．下部：タスク全体における精度．

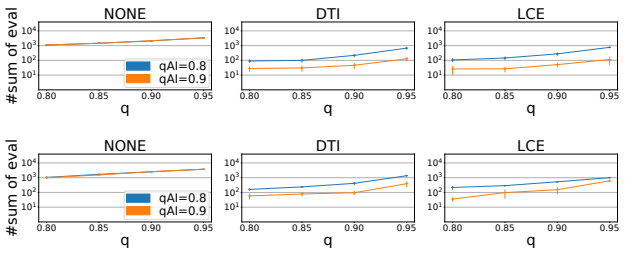

図 9　各条件でのタスククラスタ評価回数の合計．上部：Kuzushiji-MNIST，下部：Fashion-MNIST．　（縦軸は対数）

表 3 に具体的な評価回数の削減率を示す．多くの条件のもとで 70%〜90% の削減を達成しており，Kuzushiji-MNIST に対する実験の $qAI = 0.9$ の各条件においては 97% 近い削減を達成している．Kuzushiji-MNIST の $q = 0.9, qAI = 0.9$ の LCE が最も評価回数を削減しており，98.154803% の削減を達成した．最も評価回数の削減が少ない Fashion-MNIST の $q = 0.95, qAI = 0.8$ の条件の DTI でも評価回数を 63.501978% 削減した．

これらの結果は，良好な条件においては同じ計算リソースにおいて我々の手法は従来の手法と比較して数十倍から数百倍の数の AI ワーカを利用することが可能であることを示している．

表 3　各 q, qAI 条件における DTI と LCE の評価回数の削減率．同条件のフィルタなし（NONE）におけるタスククラスタ評価回数と比較して何%削減したか示している．（上段：Kuzushiji-MNIST，下段：Fashion-MNIST）

Kuzushiji-MNIST

DTI			LCE		
q	q_{AI}	削減率（%）	q	q_{AI}	削減率（%）
0.80	0.8	91.721491	0.80	0.8	90.222953
0.85	0.8	93.141365	0.85	0.8	90.069411
0.90	0.8	89.553311	0.90	0.8	86.800402
0.95	0.8	80.036673	0.95	0.8	77.434047
0.80	0.9	97.462296	0.80	0.9	97.651946
0.85	0.9	97.885280	0.85	0.9	98.154803
0.90	0.9	97.829631	0.90	0.9	97.601416
0.95	0.9	96.220280	0.95	0.9	96.614063

Fashion-MNIST

DTI			LCE		
q	q_{AI}	削減率（%）	q	q_{AI}	削減率（%）
0.80	0.8	84.682713	0.80	0.8	79.249252
0.85	0.8	85.939116	0.85	0.8	82.880154
0.90	0.8	83.553443	0.90	0.8	79.089628
0.95	0.8	63.501978	0.95	0.8	73.083787
0.80	0.9	94.256162	0.80	0.9	96.489877
0.85	0.9	94.875843	0.85	0.9	93.649238
0.90	0.9	96.262917	0.90	0.9	94.008926
0.95	0.9	89.602285	0.95	0.9	83.742331

5.3　タスククラスタの適合率

図 10，図 11 はタスク割り当て中の各イテレーションにおけるタスククラスタの適合率を示している．タスククラスタフィルタによってタスククラスタの適合率が向上していることが示された．

5.4　タスククラスタ評価回数の削減と AI ワーカに割り当てられるタスク数のトレードオフ

タスククラスタの評価回数の削減と AI ワーカに割り当てられるタスク数にはトレードオフがあると思われる．図 12 は AI ワーカに割り当てられたタスク数と評価回数を示している．多くの条件において我々の提案した手法は良いトレードオフを示している．ここでの良いトレードオフとは，「より多く AI ワーカにタスクを割り当てつつ，より多くタスククラスタの評価回数を削減する」ことである．

提案したフィルタが $q_{AI} = 0.8$ であってもタスククラスタ

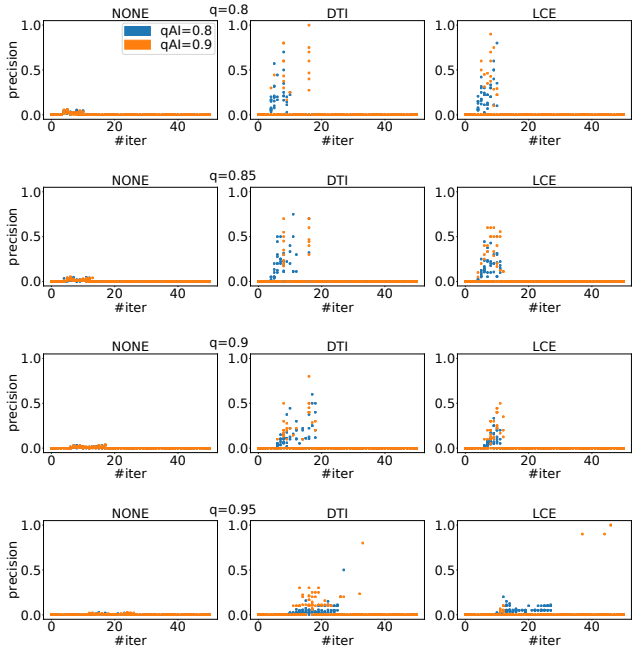

図 10　Kuzushiji-MNIST に対するタスク割り当ての各イテレーションにおけるタスククラスタの適合率．　上から順に $q = 0.8, 0.85, 0.9, 0.95$ における図．（評価がスキップされている AI ワーカのタスククラスタについても適合率は 0 としている．）

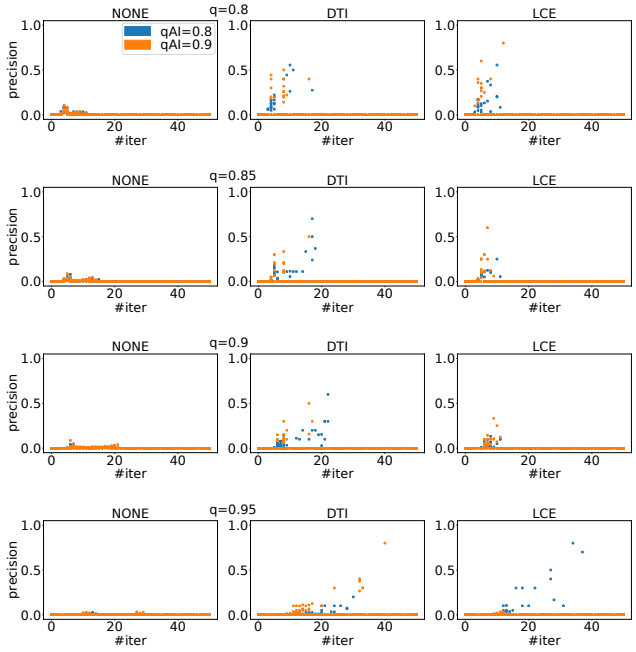

図 11　Fashion-MNIST に対するタスク割り当ての各イテレーションにおけるタスククラスタの適合率．　上から順に $q = 0.8, 0.85, 0.9, 0.95$ における図．（評価がスキップされている AI ワーカのタスククラスタについても適合率は 0 としている．）

の評価回数の大幅な削減が可能であることを考えると，特に $q_{AI} = 0.8$ において，我々の手法は AI ワーカへのタスク割り当て数を維持しながらタスククラスタの評価コストを大幅に減少させることが可能であると言える．

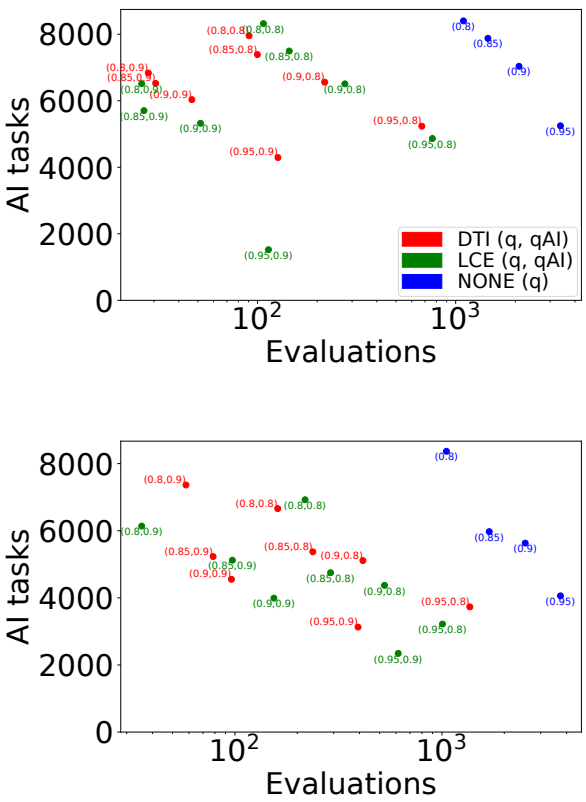

図 12　各条件における AI ワーカに割り当てられたタスク数とタスククラスタの評価回数の図．（上部：Kuzushiji-MNIST，下部：Fashion-MNIST）

DTI と LCE の両方ともタスククラスタの評価回数を大幅に減少させることが可能であるが，トレードオフに関しては若干傾向が異なる．DTI は LCE と比較して AI ワーカに割り当てるタスク数が多く，LCE は DTI と比較してタスククラスタ評価回数をより多く削減する傾向がみられる．

6　結　　論

本論文で我々はクラウド AI 開発者によって開発された AI である “AI ワーカ”と “人間ワーカ”の集団である Human+AI Crowd に対するタスク割り当ての効率化に取り組んだ．AI ワーカは人間ワーカとは異なり，スパムワーカや低品質なワーカでなくても，人間へのタスク割り当てが少なく，AI ワーカの教師データが少ないタスク割り当て初期においては低い性能を示してしまう．それゆえ，既存のタスク割り当て手法は AI ワーカの提出する結果を全て毎回評価していた．そのため，タスク割り当てにおいて AI ワーカの提出する結果を取り入れるための計算コストが大きく，スケーラビリティに難があった．

本論文では我々は評価対象の AI ワーカを減らすためのタスククラスタフィルタを提案した．また，タスククラスタフィルタの具体的な手法として Double Time-Interval Filter と Learning-Curve Estimation Filter を提案した．フィルタを用いた実験の結果は，多くの条件において我々の提案したフィル

タが AI ワーカへのタスク割り当て数をある程度維持しつつ，タスク割り当てにおける AI ワーカの評価コストを 90%以上低減することを示した．AI ワーカの評価コストの低い我々の手法を用いることで，従来の数十倍から数百倍の数の AI ワーカを利用することが可能となる．

本研究の将来的な発展として，個別の AI ワーカに特化したフィルタの開発が挙げられる．本論文では不特定多数のクラウドワーカによって開発されるブラックボックスな AI モデルの AI ワーカを想定したため，AI ワーカの具体的な実装が不明な状態でも機能するフィルタを提案した．しかし，AI ワーカの具体的な情報が得られる場合も想定される．その場合，AI ワーカごとに適切なフィルタリングを行うことでより良い結果が期待できる．もう 1 つの発展的な内容として，より高品質な AI ワーカを得るために，本論文で提案した AI ワーカの評価手法を用いることで AI ワーカに対する報酬/罰則規定の設計を行うことが考えられる．

謝　　辞

本研究の一部は JSPS 科研費 (22H00508，22K17944) の支援を受けたものである．ここに謝意を示す．

文　　献

[1] Sihem Amer-Yahia, Senjuti Basu Roy, Lei Chen, Atsuyuki Morishima, James Abello Monedero, Pierre Bourhis, François Charoy, Marina Danilevsky, Gautam Das, Gianluca Demartini, Shady Elbassuoni, David Gross-Amblard, Emilie Hoareau, Munenari Inoguchi, Jared Kenworthy, Itaru Kitahara, Dongwon Lee, Yunyao Li, Ria Mae Borromeo, Paolo Papotti, Raghav Rao, Sudeepa Roy, Pierre Senellart, Keishi Tajima, Saravanan Thirumuruganathan, Marion Tommasi, Kazutoshi Umemoto, Andrea Wiggins, and Koichiro Yoshida. Making ai machines work for humans in fow. *SIGMOD Rec.*, Vol. 49, No. 2, p. 30–35, dec 2020.

[2] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *CoRR*, Vol. abs/1812.01718, , 2018.

[3] David Gross-Amblard, Atsuyuki Morishima, Saravanan Thirumuruganathan, Marion Tommasi, and Ko Yoshida. Platform Design for Crowdsourcing and Future of Work. *Bulletin of the Technical Committee on Data Engineering*, Vol. 42, No. 4, March 2019.

[4] Masafumi Hayashi, Masaki Kobayashi, Masaki Matsubara, Toshiyuki Amagasa, and Atsuyuki Morishima. Incentive design for crowdsourced development of selective ai for human and machine data processing: A case study. In *2019 IEEE International Conference on Big Data (Big Data)*, pp. 4596–4601, 2019.

[5] Tomoya Kanda, Hiroyoshi Ito, and Atsuyuki Morishima. Efficient evaluation of ai workers for the human+ai crowd task assignment. In *The 6th IEEE Workshop on Human-in-the-Loop Methods and Future of Work in Big Data (HMData 2022), In Proceedings of the 2022 IEEE International Conference on Big Data*, pp. 3985–3991, Dec. 2022.

[6] Masaki Kobayashi, Kei Wakabayashi, and Atsuyuki Morishima. Human+ai crowd task assignment considering result quality requirements. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 9, No. 1, pp. 97–107, Oct. 2021.

[7] Prasanth Kolachina, Nicola Cancedda, Marc Dymetman, and Sriram Venkatapathy. Prediction of learning curves in machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22–30, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[8] Felix Mohr and Jan N. van Rijn. Learning curves for decision making in supervised machine learning - A survey. *CoRR*, Vol. abs/2201.12150, , 2022.

[9] Aaron N. Richter and Taghi M. Khoshgoftaar. Approximating learning curves for imbalanced big data with limited labels. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 237–242, 2019.

[10] Tom J. Viering and Marco Loog. The shape of learning curves: a review. *CoRR*, Vol. abs/2103.10948, , 2021.

[11] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. 2017.

付　　録

Appendix A: 実験で用いた AI ワーカの一覧

各 AI ワーカ名は scikit-learn[10]のクラス名を指している．

- MLPClassifier
- ExtraTreeClassifier
- LogisticRegression
- KMeans
- DecisionTreeClassifier
- SVC
- KNeighborsClassifier
- GaussianProcessClassifier
- MultinomialNB
- AdaBoostClassifier
- PassiveAggressiveClassifier
- RidgeClassifier
- RidgeClassifierCV
- ComplementNB
- NearestCentroid

Appendix B: 実 験 環 境

AMD Ryzen 9 5900X 12-Core Processor, 64GB RAM, GeForce RTX 3090 GPU, Ubuntu 20.04.4, Python 3.8.10, Scikit-learn 0.24.2

10：https://scikit-learn.org/stable/