Vol.2020-HCI-186 No.36
2020/1/16

クラウドソーシングにおける動的な回答収集による低コストな多数決手法

森永 聖也[1,a)]　斎藤 奨[1]　中野 鐵兵[1]　小林 哲則[1]　小川 哲司[1]

概要：二値ラベリングのタスクを題材とし，更なる回答収集が必要と判断されたデータに対してのみ動的に回答を追加収集することにより，従来通りのラベル品質を保ちながらより安価に多数決を行う手法を提案する．クラウドソーシングにおける出力の品質向上のためには，1 データあたり複数の回答を募り多数決を行う方法がしばしば用いられている．あらかじめ十分な数の回答が集まっていれば出力精度は大きく改善される一方，収集回答数に比例し金銭コストも増大するという問題がある．そこで本研究では（1）現時点で収集済みの回答を用いて多数決を行い（2）曖昧なデータは収集した回答数が足りないものと仮定し該当データのみ追加で回答を収集する．という手続きを収集回答数が収束するまで繰り返す手法を提案する．事前に用意したラベリングタスクを用いて評価実験を行った結果，全てのデータに対して同数の十分な回答を収集した場合と同等もしくはそれ以上の精度を達成しながら，最大で 28%のコスト削減に成功した．

キーワード：クラウドソーシング，品質管理，コスト管理

1. まえがき

機械学習のためのデータ収集においてクラウドソーシングが広く利用されている [1], [2], [3]．インターネットを通じて不特定多数の人々（ワーカー）へ作業を依頼することへの利便性が高く評価されている一方，悪意のあるワーカー回答などに起因するデータ品質の劣化が一般的な問題として知られている．手軽なデータ品質保証の手法としては 1 データあたり複数の回答を募り多数決を行う方法が挙げられる [4], [5], [6]．あらかじめ十分な数の回答を集めることで回答品質は大きく改善される [7] が，収集する回答数に比例して金銭コストも増大する．一方，回答数が少なすぎると必要な予算を削減できるが，誤ったラベリングが行われる可能性が高くなる．こうしたトレードオフを踏まえ，タスク依頼者（リクエスター）は想定する予算を考慮しながら適切に収集する回答数を設定する事が必要となる．しかし，未知のデータについて収集を行う際にリクエスターが適切な収集回答数を設定することは容易ではない．

データ品質保証を低コストに実現する様々な先行研究があるが，その多くはシステムが任意のワーカーにタスクを依頼し，依頼したワーカーによって回答が行われるという前提に基づいている．Karger らは確率伝搬法と低ランク行列近似を用いて，どのタスクをどのワーカーに割り当てるか決定し，ワーカーの回答から正解を推測するアルゴリズムを提案している [8]．また Bachrach らはデータの難しさ，ワーカの能力及びデータの正解を推定するグラフィカルモデルを用いてタスクの割り当てを行っている [9]．Yan らは能動学習によって回答が信頼できるワーカーを選択する手法が提案されている [10]．Basu らはクラウドソーシングによってニュース記事等の作成を行うためのタスクのワーカーへの割り当てを最適化する機能を持ったプラットフォームを開発している [11]．いずれの手法も，ワーカーが信頼できるか判断し，信頼できるワーカーにタスクを依頼することで，データの判断に必要な回答数を少なくし，低コスト化を実現する．しかし既存のプラットフォーム（e.g., Amazon Mechanical Turk, AMT）の多くでは回答待ちのタスク一覧からワーカーがタスクを選択するという方式が一般的であるため，システムがワーカーを選択してタスクを割り当てる手法を採用することができない．

そこで本研究では，プラットフォームからのタスク割り当てを前提としない低コストなデータ品質保証の手法として，回答が曖昧なデータについてのみ動的に回答を追加収集することで，ラベルの品質保証を行いながら低コストにデータの収集を行う枠組みを提案する．二値ラベリングのタスクを題材とし，正解ラベルが付与されたサブセットに対して動的な回答収集のシミュレーションを行うことでど

[1] 早稲田大学
Waseda University
[a)] morinaga@pcl.cs.waseda.ac.jp

のような方式でラベリングを行うのが最も効率的かを決定する．得られた結果からデータ毎に共通して収集すべき回答数などのパラメータを決定し，未知データ全体のデータ収集に用いる．動的な回答の収集の流れとしては，(1) 現時点で収集済みの回答を用いて多数決を行い (2) 曖昧なデータは収集する回答数が足りないものと仮定し該当データのみ追加で回答収集する．という手続きを，収集する回答数が収束するまで繰り返す．この手法では正解ラベルが付与されたサブセットが必須であり，収集する回答数等パラメータの推定はサブセットを用いて初めて可能となる．しかしながら機械学習で用いるラベル付与をしたいデータセットの多くには，学習用データセット等多少なりとも予めラベルが付与されたデータのサブセットの存在を仮定することができる．またクラウドソーシングにおいてワーカーのパフォーマンス評価を適切に行う際に，一部に正解が既知なデータを紛れ込ませる方法が用いられることも多く [12][13]，サブセットを用意することは必ずしもリクエスタにとって余分な労力とはならない．本手法はこのサブセットを用いることで全体として収集する回答数を抑え，低コストで高品質なデータ取得を可能とすることを目指す．

2. 動的な回答収集による多数決手法

2.1 基本アプローチ

提案手法ではラベリング対象の全タスクに対して予め定めた人数のワーカーにラベリング作業を発注し，予め定めた多数決の基準に達しなかったタスクに対してのみ動的に発注を繰り返す．予め定めた人数が低ければ低いほどコスト削減効果が期待できる一方，悪意のあるワーカーやタスクを正しく理解していないワーカーの影響から品質保証が困難となる．同様に繰り返し条件が厳しければより高い品質が期待できる一方，コストもその分増大する．タスクが容易であれば少ない人数で高い品質の結果を期待できるが，ワーカーに対するタスクの難易度の推定も容易ではない．正確なラベリングに必要な回答数が分からないためリクエスターが直感的に収集する回答数を決めるなどした結果，多数決の結果から期待した品質が得られなかったという事態が起きてしまう．

本手法では，低コストで高品質なデータ取得を可能にする動的な回答収集に必要なパラメータを，対象データのサブセットを用いて自動的に推定する枠組みを提案する．具体的には，まず正解ラベルが既知となっている対象データのサブセットに対して事前にクラウドソーシングで回答収集する．次に収集データに対して必要なパラメータ (e.g., 1 データあたりの収集回答数，追加で収集するデータを判定する条件，スパマーと判定する閾値) を変更しながらシミュレーションを行い，対象のデータ収集において適切なパラメータを決定する．最後に決定したパラメータを用いて動的な回答収集を実施し，多数決によりラベルを決定する．

Algorithm 1 パラメータ決定アルゴリズム

Input: ラベリング対象のデータセット T の任意サブセット T'
Output: T をラベリングする際のパラメータと評価

初期化
if (T' に正解ラベルが付与されていない) **then**
  T' に正解ラベルを付与
end if
$N_{max} \Leftarrow 1$ データあたりの収集回答数の最大値
$A_{max} \Leftarrow$ ラベル正解精度の目標値
$L \Leftarrow$ クラウドソーシングで取得した T' のラベル
for $n = 3, n <= N_{max}, n+ = 2$ **do**
  合計 n となるラベルを収集し L へ追加
  $a \Leftarrow L$ の多数決による正解精度
  if ($a > A_{max}$) **then**
    $N_{max} \Leftarrow n$; break
  end if
end for
パラメータ決定のための繰り返し処理
$\Omega \Leftarrow$ パラメータ集合 ω の集合
$\Phi \Leftarrow \emptyset$
for $n = 2$ to N_{max} **do**
  for ω in Ω **do**
    $L'_n \Leftarrow$ 回答数 n だけ取り出した L のサブセット
    while (True) **do**
      $l \Leftarrow$ 繰返し条件 ω に従って L' から選択されたラベル
      if ($l \notin \emptyset$) **then**
        $L'_n \Leftarrow l$ を追加
      else
        $a_\omega \Leftarrow L'$ の多数決条件 ω 時の正解精度
        $c_\omega \Leftarrow L'$ の多数決条件 ω 時の要素数
        $score_\omega \Leftarrow$ 評価関数 $E(a_\omega, c_\omega)$
        break
      end if
    end while
  end for
  $\omega_n \Leftarrow \arg\max_\omega score_\omega$
  $\phi_n \Leftarrow (\omega_n, a_{\omega_n}, c_{\omega_n}, score_{\omega_n})$
  初期回答数 n 時の最適パラメータ集合 ϕ_n を Φ に追加
end for
return Φ

2.2 パラメータ決定アルゴリズム

動的な回答収集に必要なパラメータ決定のための具体的な手順を Algorithm 1 に示す．

ここではまず，ラベリング対象のデータセット T のサブセット T' をパラメータ推定用評価データとして選択する．事前に正解ラベルが与えられているサブセットがある場合はそれを T' として利用し，ない場合は全体の 5%等条件を決めてランダムにデータを選択し，クラウドソーシング以外の信頼できる手法で正解ラベルを与える．次に T' に対して，1 データ当たり最大 N_{max} の回答をクラウドソーシングでデータ収集を行う．多数決を前提とするため $N_{max} \geq 3$ である．事前に目標精度 A_{max} を定めておき，1 データ当たりの回答数を 3 から増やしながら目標精度 A_{max} を達成する回答数を N_{max} としても良い．いずれの場合も T' に含まれるタスク数を $N_{T'}$ とした時，クラウドソーシングで集められたラベルの集合 L の要素数は $N_{T'} \times N_{max}$ とな

る．(Algorithm 1[1–13])

さらに動的な回答収集を行うためのパラメータ集合 ω の集合 Ω を設計する．パラメータ集合 ω は追加でラベルを収集する対象のデータか否かを判定する条件に加え，採用する多数決のアルゴリズムに応じて必要な様々なパラメータが含まれる．例えば単純多数決を採用する場合には投票率閾値 t をパラメータとして用意し，得票率 x が $t < x < 1-t$ となったデータを追加でラベルを取得するデータとする．また，例えばスパマー除去を用いる場合にはスパマーと判定するスコアの閾値をパラメータとして用意する．(Algorithm 1[14])

次にラベルの集合 L を用いて初期回答数を n として動的な回答収集を行った場合の最適なパラメータの集合 ω_n を求める．初期回答数 n によって最適な ω が異なる場合に備え，最適なパラメータの集合 ω_n を n 毎に求める必要がある．そのためにまずパラメータ集合 ω を決め，L から 1 データ当たりの回答数を n としたサブセット L' を抽出する．L' は要素数 $N_{T'} \times n$ のワーカーから得られたラベルの集合である．次に ω のうち多数決に関するパラメータを用いて L' から追加でラベルを取得するデータを表す集合 l を求める．L' に l の集合を加え l が空集合となるまで一連の処理を繰り返す．この繰返し処理はワーカーから得られた回答が曖昧なデータを選択し，該当データのみ追加で回答を収集する処理のシミュレーションに相当する．ω 毎に回答の評価値を計算し，最も良い評価値を与える ω_n，及びその時の正解精度と総回答者数，評価値を Φ_n として保存する．なおより厳密なシミュレーションとするためにはサブセット L' を複数組み合わせで抽出し，統計的に評価して ω_n を求めることが望ましい．最後に全ての初期回答数 n に対応するパラメータと評価の集合 ϕ_n の集合 Φ を返し，費用対効果の観点から初期回答数 n とその場合の最適なパラメータ ω_n が決定される．(Algorithm 1[15-35])

3. データ収集

提案手法が実タスクにおいて動作するか検証するために，画像に対するラベル付けタスクを設計し，実際に AMT を用いて画像に対するラベル収集を行った．

3.1 クラウドソーシングに向けたタスク設計

クラウドソーシングを用いて回答を収集するために，専門的な知識が無いワーカーにも回答が可能な様に設計された実タスクを用いた．対象の画像は映像データから，1 秒につき 1 枚ずつ画像として切り出し，YOLOv2[14] によって検出した牛の領域画像である．提示された画像について，牛の特定部位が含まれた画像であれば”Yes”，含まれていなければ”No”のラベルを付与する．また，YOLOv2 の検出した画像の中に人や建物の一部など，牛以外を誤検知している画像に備え，牛が写っていない時，人が写っている時にはそれぞれ，”No cows”，”Exist person”のラベルを付与する．誤検知している画像については，収集後に著者らで確認し，牛以外が写っている画像は実験に利用したデータには含まれていない．専門的な知識が無いワーカーにも回答が可能となる様に，タスクでは回答前にイントロダクションとして具体的な画像例を提示した．またタスクの回答画面には，イントロダクションを再度確認するボタンと問題文，問いたい画像，ラベル付与のボタンを配置した．本タスクは 1 セット 20 問とし，1 セット回答する毎に報酬が支払われる．実際のタスクで用いた回答画面を図 1 に示す．

図 1　実際のタスクの回答画面

表 1　ワーカーの回答の内訳

ラベル	回答数
Yes	18458
No	69883
No cows	581
Exist person	102

3.2 アノテーション

3.1 で構築したタスクを，AMT に発注し，実際のワーカーにアノテーションを依頼した．牛以外が写された画像を除いた際のタスク全体のデータ数は 6848 枚である．1 データ辺り 13 人に依頼し，タスクに回答したワーカーは合計 733 人，回答数の合計は最大 6848 × 13 回答となる．回答者のラベルの内訳を表 1 に示す．

また，6848 枚の画像に対して，著者らが特定部位の有無についての正解ラベルを手動で与えた．付与したラベルの内訳は，特定部位を含む画像が 1281 枚，含まない画像が 5567 枚である．

4. 動的な回答収集の効果検証実験

第 2 節で提案した手法及び第 3 節述べたデータセットを用いて，動的な回答収集の効果がどの程度現れるかを検証する実験を行った．提案手法ではデータセットのサブセットを用いてパラメータの決定を行うとしているが，本実験ではサブセットではなく全データに正解ラベルが付与されているデータセットを用いてその効果を検証する．

4.1 多数決手法

収集した回答からラベルを決定する手法によらず提案手法が機能することを確認するために，下記 3 通りの手法でシミュレーションを行う．

単純多数決 収集した回答のうち最も得票数の多いラベルを採用する．パラメータは得票率の閾値 t とし，少数派の回答の割合が閾値 t を超えている場合に追加でラベルを収集する．追加で収集する回答数が収束した時点で収集を終了する．

スパマー除去後の単純多数決 スパマー検知のための指標であるスパマースコア [15] を用いたスパマー除去を行い，残った回答のうち最も得票数の多いラベルを採用する．パラメータは得票率の閾値 t，スパマースコアの閾値 s，及び終了条件 f とする．閾値 t は単純多数決と同様に追加でラベルを収集する閾値として用いる．スパマースコアはワーカの能力から算出され，0 に近いほどスパマーであることが疑われる数値であり，ここではスパマースコアが一定値 s 未満であるワーカーをスパマーと判定し除去する．回答の収集を終了する条件 f は，単純多数決と同様の条件である 1. 追加で収集する回答数の収束と，2. 回答したワーカーに含まれるスパマーの収束のいずれかとする．

重み付き多数決 ワーカーの能力を考慮した重み付き多数決 [16] の結果得られたラベルを採用する．スパマー除去後の単純多数決と同様に，スパマースコアを用いてスパマーの判定を行い，データに存在するスパマーの数だけ，追加でラベルを収集する．パラメータはスパマー除去後の単純多数決と同様で得票率の閾値 t，スパマースコアの閾値 s，及び終了条件 f とする．

4.2 実験設定

シミュレーションを行う際の実験設定を表 2 に示す．項目は第 2 節の Algorithm 1 の変数名に対応している．この設定に基づいて，多数決手法毎にシミュレーションを行いその結果を比較する．

初期回答数 n は 2 から開始し 10 まで繰り返すものとし，パラメータ t，s，回答の収集を終了する条件を変動させて最良となる組み合わせを選択する．n 個の回答を選択するワーカーの順番はランダムとし，収集した L の中から無作為に L' 抽出した．L' は 1 つの n につき 3 回抽出し最適な ω_n は評価値の平均値が最も良くなるパラメータ ω とした．評価関数 E として，正解精度及びコストから費用対効果を示す指標として定義した E を利用した．正解精度としては $n = 3$ における繰り返しを行わない場合の精度を基準として計算したエラー改善率を用いた．コストとしては $n = N_{max}(= 13)$ の時の総回答数を基準としたコストの相対値を用いた．

表 2 実験設定

項目	内容
T'	牛画像データセット 6848 枚
N_{max}	13 人
L	6848 × 13 回答
初期回答数 n	[2,3,4,5,6,7,8,9,10]
得票率の閾値 t	[0.05,0.1,0.15,0.2,0.3]
スパマースコアの閾値 s	[0.05,0.1,0.15,0.2,0.3]
評価関数 E	（改善率）／（コスト）

4.3 実験結果

単純多数決，スパマー除去を行う単純多数決，重み付き多数決を用いた場合の精度と総回答数の推移の例を，それぞれ図 2，図 3，図 4 に示す．ただし，単純多数決及びスパマー除去を行う単純多数決について，n の初期値が偶数のときは，$2k + 1$ 回目の結果を，n の初期値が奇数の時は，$2k$ 回目の結果を用いている（$k = 0, 1, 2...$）．n の初期値が偶数の時の，$2k$ 回目の結果，n の初期値が奇数の時の，$2k + 1$ 回目の結果では，データあたりの回答数の最大は偶数となる．従って，Yes，No，の回答数が同数となるデータが存在し，算出される精度がタスク全体の精度ではないため用いなかった．また実験結果を表にしたものを表 3 に示す．表は 3 つの手法について，n と繰り返しの有無を変動させた結果を示したものであり，改善率はそれぞれの手法での $n = 3$ における繰り返しを行わない場合の精度を基準として評価した．

実験結果から，スパマー除去後の単純多数決において，$n = 3$ を初期値とした場合が最も E の値が高くなった．従って本タスクでは，スパマー除去後の単純多数決において，$n = 3$ を初期値としてスパマー除去後の単純多数決を行うことで，データ収集の費用対効果が最も高くなると考えられる．ここで，$n = 3$ を初期値としてスパマー除去後の単純多数決について，データ毎の回答数を図 5 に示す．横軸はデータあたりの回答数，縦軸はデータの通し番号である．この時，グラフの外枠の面積は，全てのデータについて $N_{max} = 13$ 回答ずつ収集した場合の総回答数であり，金銭コストと見做す事が出来る．従って，内部の青い部分の面積は，$n = 3$ を初期値としてスパマー除去後の単純多数決を行った場合の金銭コストとなり，外枠との面積を比べる事で金銭コストが削減されている事が確認できる．

また，いずれの多数決手法においても動的な回答収集をする手法の有効性が確認できる．例えば，$n = 3$ を初期値として，繰り返しスパマー除去後の単純多数決を行った場合の精度は，$n = 9$ の時，繰り返しを行わなかった場合の精度を上回っており，金銭コストについては，繰り返しを行わない場合の 59%に抑えられた．特に単純多数決の結果を見ると，n が 9 以上の繰返しを行う場合では $n = N_{max} = 13$ で繰返しを行わない場合と同等以上の精度となっており，最大 28%のコスト削減に成功している．他の手法について

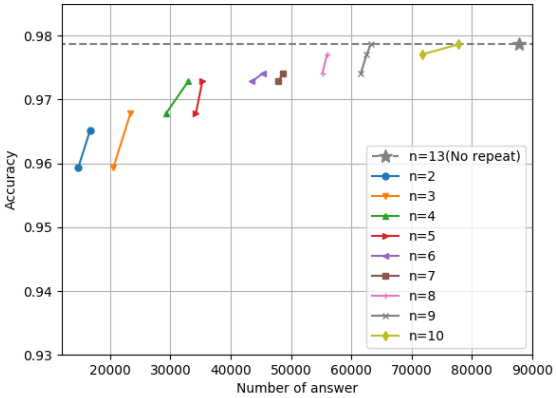

図 2 単純多数決を用いて n を変動させた時の精度の推移例

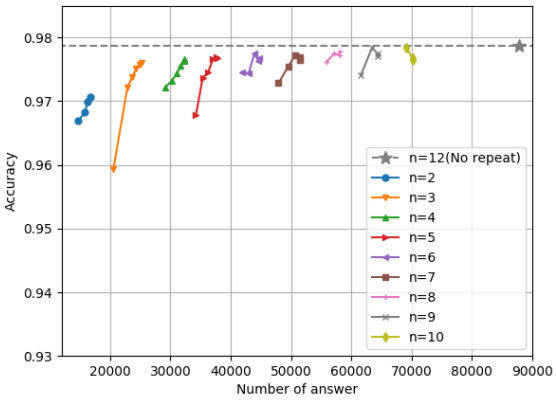

図 3 スパマー除去を行う単純多数決を用いて n を変動させた時の精度の推移例

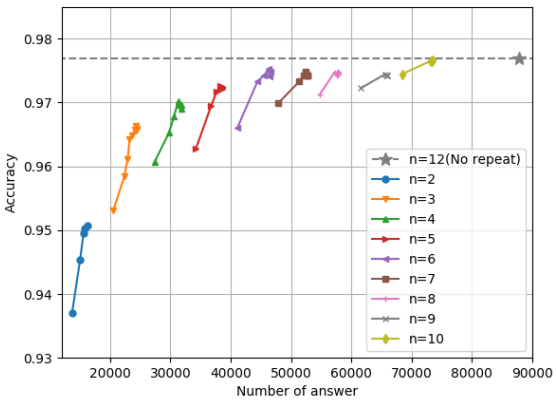

図 4 重み付き多数決を用いて n を変動させた際の精度の推移例

も，同様に有効性が確認できる．

単純多数決及びスパマー除去後の単純多数決の結果を比較すると，スパマー除去を用いることでより少ないワーカー数で上限値に近づく傾向がみられる．$n = 3$ を初期値として，繰り返しスパマー除去後の単純多数決を行った場合の精度は，$n = 7$ を初期値として，繰り返し単純多数決を行った場合よりも精度が高い．従って，スパマー除去は低コストに，高品質なデータを収集するために有効であると考えられる．

一方で，重み付き多数決の結果を見ると，他 2 つの手法と比べて全体的に精度が低い傾向があり，ワーカーの能力の推定が有効に機能していないと考えられる．スパマー除去後の単純多数決の結果が最も精度が高いことから，ワーカーの能力によるスパマーの検知は機能していると考えら

表 3 最終的な精度及びコスト：(上段) 単純多数決 (中段) スパマー除去後の単純多数決 (下段) 重み付き多数決

n	繰返し	精度	総回答数	改善率	コスト	E
2	有	0.965	16686	0.129	0.190	0.677
3	無	0.960	20542	0.000	0.234	0.000
3	有	0.969	22657	0.236	0.258	0.916
4	有	0.972	32941	0.316	0.375	0.841
5	無	0.968	34226	0.197	0.390	0.505
5	有	0.972	35304	0.307	0.402	0.763
6	有	0.975	45382	0.379	0.517	0.733
7	無	0.972	47906	0.307	0.546	0.562
7	有	0.975	48603	0.384	0.554	0.693
8	有	0.977	55954	0.440	0.637	0.691
9	無	0.975	61578	0.384	0.701	0.547
9	有	0.979	63203	0.472	0.720	0.656
10	有	0.979	77879	0.472	0.887	0.532
13	無	0.979	87781	0.472	1.000	0.472
2	有	0.970	17016	0.255	0.194	1.314
3	無	0.960	20542	0.000	0.234	0.000
3	有	0.976	25385	0.417	0.289	1.441
4	有	0.975	31718	0.395	0.361	1.092
5	無	0.968	34226	0.197	0.390	0.505
5	有	0.976	37712	0.415	0.430	0.966
6	有	0.977	44271	0.433	0.504	0.859
7	無	0.972	47906	0.307	0.546	0.562
7	有	0.977	51498	0.443	0.587	0.755
8	有	0.978	58027	0.445	0.661	0.674
9	無	0.975	61578	0.384	0.701	0.547
9	有	0.977	64414	0.442	0.734	0.602
10	有	0.977	70194	0.429	0.800	0.536
13	無	0.979	87781	0.472	1.000	0.472
2	無	0.939	13695	-0.278	0.156	-1.784
2	有	0.950	16174	-0.044	0.184	-0.240
3	無	0.952	20542	0.000	0.234	0.000
3	有	0.965	24629	0.264	0.281	0.941
4	無	0.958	27384	0.127	0.312	0.406
4	有	0.970	32095	0.377	0.366	1.031
5	無	0.962	34226	0.198	0.390	0.508
5	有	0.971	38588	0.389	0.440	0.886
6	無	0.965	41067	0.275	0.468	0.587
6	有	0.975	46701	0.469	0.532	0.881
7	無	0.969	47906	0.358	0.546	0.656
7	有	0.974	52704	0.466	0.600	0.776
8	無	0.971	54743	0.388	0.624	0.622
8	有	0.975	57978	0.479	0.660	0.725
9	無	0.972	61578	0.410	0.701	0.584
9	有	0.975	65974	0.482	0.752	0.641
10	無	0.974	68405	0.464	0.779	0.596
10	有	0.976	73550	0.507	0.838	0.605
13	無	0.977	87781	0.518	1.000	0.518

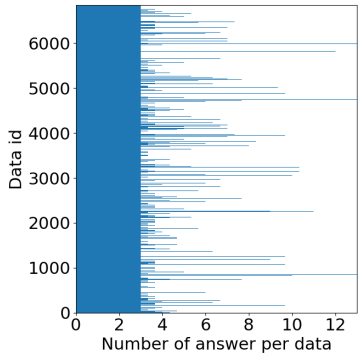

図 5 $n = 3$ を初期値としたスパマー除去後の単純多数決におけるデータ毎の回答数

れる．従って，能力が高いワーカーを過小評価しているのではないかと推測される．

5. まとめ

本研究では，低コストなデータ品質保証の手法として，データ毎にラベルの推定に必要な数だけの回答を動的に収集し，ラベルの品質保証を行いながら低コストにデータの収集を行う枠組みを提案し，効果を検証した．事前に用意したラベリングタスクを用いて評価実験を行った結果，全てのデータに対して同数の十分な回答を収集した場合と，同等以上の精度を達成しながら，最大 28%のコスト削減に成功した．一方で，本研究では実験に使用したデータセットが 1 種類であることから，タスクに依存せずに，同様の結果がいえるか明らかでない．まず，他のタスクにおいても提案手法が有効か確認を行い，その後，開発中のクラウドソーシングのプラットフォームに，提案した枠組みを組み込み，公開することを検討している．

参考文献

[1] Deng, J. et al.: ImageNet: A large-scale hierarchical image database, *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255 (2009).
[2] Patterson, G. et al.: SUN attribute database: Discovering, annotating, and recognizing scene attributes, *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2751–2758 (2012).
[3] Chen et al.: Collecting Highly Parallel Data for Paraphrase Evaluation, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, Association for Computational Linguistics, pp. 190–200 (2011).
[4] Snow et al.: Cheap and Fast—but is It Good?: Evaluating Non-expert Annotations for Natural Language Tasks, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, Stroudsburg, PA, USA, Association for Computational Linguistics, pp. 254–263 (2008).
[5] Sorokin, A. et al.: Utility data annotation with Amazon Mechanical Turk, *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8 (2008).
[6] Callison-Burch et al.: Fast, Cheap, and Creative: Evaluating Translation Quality Using Amazon's Mechanical Turk, *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, Association for Computational Linguistics, pp. 286–295 (2009).
[7] Zheng et al.: Truth Inference in Crowdsourcing: Is the Problem Solved?, *Proc. VLDB Endow.*, Vol. 10, No. 5, pp. 541–552 (2017).
[8] Karger, D. R. et al.: Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems (2011).
[9] Bachrach, Y. et al.: How To Grade a Test Without Knowing the Answers — A Bayesian Graphical Model for Adaptive Crowdsourcing and Aptitude Testing (2012).
[10] Yan et al.: Active Learning from Crowds, *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, USA, Omnipress, pp. 1161–1168 (2011).
[11] Roy, B. et al.: Task Assignment Optimization in Knowledge-intensive Crowdsourcing, *The VLDB Journal*, Vol. 24, No. 4, pp. 467–491 (2015).
[12] Kittur et al.: Crowdsourcing User Studies with Mechanical Turk, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, New York, NY, USA, ACM, pp. 453–456 (2008).
[13] Huang et al.: Enhancing Reliability Using Peer Consistency Evaluation in Human Computation, *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, New York, NY, USA, ACM, pp. 639–648 (2013).
[14] Redmon, J. et al.: YOLO9000: Better, Faster, Stronger (2016).
[15] Raykar, V. C. et al.: Ranking annotators for crowdsourced labeling tasks, *Advances in Neural Information Processing Systems 24* (Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F. and Weinberger, K. Q., eds.), Curran Associates, Inc., pp. 1809–1817 (2011).
[16] Dawid, P. et al.: Maximum likelihood estimation of observer error-rates using the EM algorithm, *Applied Statistics*, pp. 20–28 (1979).