DEIM2020 I5-2(day2 p70)

クラウドソーシングにおける多数決の精度向上のためのワーカー組合せの選出手法

松田　浩幸†　　田島　敬史†

† 京都大学大学院情報学研究科　〒 606–8501 京都府京都市左京区吉田本町
E-mail: †matsuda-hiroyuki@dl.soc.i.kyoto-u.ac.jp, ††tajima@i.kyoto-u.ac.jp

あらまし　本研究では，クラウドソーシングにおいて，全ワーカーの中から多数決の精度が高くなるようなワーカーの組合せを選出する手法を提案する．多数決が失敗する状況として考えられるのは，一つの不正解の選択肢にワーカーの回答が偏ってしまうような場面である．できるだけこのような状況を回避するためには，各ワーカーの回答が一つの不正解の選択肢に偏る確率が低くなるようなワーカーの組合せを選出することが重要である．そこで，各ワーカーの回答データをベクトル化し，クラスタリングを行う．クラスタリングの結果，各クラスタは間違い方の傾向の似通ったワーカーで構成される．異なるクラスタのワーカーを組み合わせることで，多数決の精度の高いワーカーの組合せを選出する．実験の結果，回答の傾向が異なるワーカーが存在するような場合に，多数決の精度の高いワーカーの組合せを選出することができた．

キーワード　クラウドソーシング，多数決，品質管理

1　はじめに

クラウドソーシングとは，インターネットなどを介して不特定多数の人々に作業を依頼する仕組みのことである．例えば，文章の校正をするタスクや，Web サイトの感想を入力するタスクなど比較的安価なコストで様々なタスクを依頼することができる．また，画像などのアノテーションはこれまで少数の専門家によって行われることが多かったが，クラウドソーシングによって専門家ではない不特定多数のワーカーによっても行われるようになった [2]．これにより，機械学習などに用いる大量の正解データなど，計算機では生成するのが困難であったデータを低コストで収集することが可能になった．そして，近年ではクラウドソーシングをサポートする様々なサービスが広く用いられるようになってきており， Amazon によって開始された Amazon Mechanical Turk [1] などのクラウドソーシングのプラットフォームによって市場は急速に拡大している．

しかしながら，従来の比較的コントロールされた環境下において取得されたデータと比較すると，クラウドソーシングのワーカーは能力や意欲にばらつきがあるために，得られるデータの品質にばらつきが生じる．また，報酬目的で全てのタスクにランダムな回答をするなど，悪質なワーカーが混じってしまうため，品質管理はクラウドソーシングにおける重要なテーマの一つである．安定して高品質な結果を得るための仕組みとして広く採用されているのが，単一のタスクを複数のワーカーに依頼して冗長性を上げ，その回答を統合して信頼性を向上させるという手法である．クラウドソーシングで依頼される典型的な作業の一つに，データへのラベル付けなど，ある問題に対して複数の選択肢の中から正解だと思われる選択肢を回答させる多肢選択問題があるが，多肢選択問題に対しては多数決が有用な統合手法の一つとして挙げられる．

複数のワーカーに単一のタスクを依頼して多数決によって統合するとき，どのワーカーを選出するかは重要である．テスト問題の正解率が上位の複数のワーカーを選出してタスクを依頼すれば，ある程度の品質の結果が得られると考えられるが，高品質な結果が求められるようなタスクにおいては，単純な選出手法よりも精度が高くなるような選出手法が必要である．

本研究では，全ワーカーの中から単一のタスクでの多数決の精度が高くなるようなワーカーの組合せを選出する手法を提案する．多数決が失敗してしまう状況として考えられるのは，一つの不正解の選択肢にワーカーの回答が偏ってしまうような場面である．できるだけこのような状況を回避するためには，選出した各ワーカーの回答が不正解である一つの選択肢に偏る確率が低くなるようなワーカーの組合せを選出することが重要である．各ワーカーの中には回答の傾向が似通ったワーカーも存在するので，テスト問題に対する正解率が高かったとしても，回答の傾向が近いワーカー同士は選出するべきではないと考えられる．そこで，まず，各ワーカーの過去の回答データからワーカーの特徴ベクトルを定義する．次に，そのベクトルを用いてワーカーをクラスタリングする．その結果各クラスタは間違い方の傾向の似通ったワーカーによって構成される．つまり，同クラスタ内のワーカーを組み合わせてしまうと，多数決の精度があまり向上しないと考えられる．最後に，異なるクラスタからワーカーを組み合わせることで，多数決の精度の高いワーカーの組合せを選出する．

本論文の構成は以下の通りである．第 2 章において，本研究と関連のあるクラウドソーシングの統計的な品質管理手法に関する研究など，提案手法に関連した研究について述べる．第 3 章では，本研究で扱う問題の設定について説明する．第 4 章では，多数決が正解する期待値の計算方法について説明する．そ

の次に，愚直な手法と提案手法の説明を行い，第 7 章において提案手法の性能評価のための計算機実験について述べる．第 8 章では，本論文の結論を述べるとともに，今後の課題について説明する．

2 関連研究

クラウドソーシングにおける品質管理については様々な研究が行われている．不特定多数のワーカーの回答から真の正解を推定するための最も単純な手法は，同じ問題に対して複数のワーカーから回答を得て多数決を取ることである．しかし，単純な多数決では能力の高いワーカーと同じ選択肢を選び続けるような悪質なワーカーの回答の重みが変わらないので，多数決をしてもあまり品質が向上しない．そこで，正解率によって各ワーカーの回答の重みを調整したり，割り当てるタスクを増減させるような手法が提案されている．

Dawid ら [3] は，信頼度の異なる複数の医師が複数の患者を診断した結果から正しい診断結果を推定する，EM アルゴリズムを利用した手法を提案した．この研究は医療の診断における文脈で行われたものであったが，クラウドソーシングの品質管理の文脈においても様々な研究の基礎となっており，Dawid らの手法を拡張するような研究も盛んに行われている．例えば，各ワーカの性能と各タスクの難易度を考慮した手法 [9] や，ワーカの性能とタスクの難易度を多次元化した手法 [8]，正解率は低くてもバイアスが一貫しているようなワーカーを生かす手法 [4]，などが挙げられる．また，Joglekar ら [5] [6] は，多数決における回答の一致を用いた，ワーカの誤り率を推定する手法を提案した．

また，Wu ら [10] は，クラウドソーシングにおいて単一のタスクを複数のワーカーに依頼するときに，多様性を重視したワーカーを選出する手法を提案した．各ワーカーの類似度をプロフィールの一致度や過去の異なるタスクにおける回答データなどから計算し，多様性の度合いとして扱う．そして，ワーカー同士の類似度の平均が低い，つまり多様性の高いようなワーカーの組み合わせを選出する．この手法はワーカーの類似度を計算し，類似度の低いワーカーを選出するという点で提案手法に近い手法であるが，提案手法ではワーカーの回答の中で誤答が被ってしまった部分のみを用いて類似度を計算しているという点で異なる．

3 問題設定

3.1 文字の設定と研究の目的

本研究で扱う全ワーカーの中から多数決の精度が高くなるようなワーカーの組合せを選出する問題について定義する．まず，ワーカーが取り組むタスクとして，典型的なクラウドソーシングのタスクである多肢選択問題を用いる．多肢選択問題では，ある問題に対して複数の選択肢の中から正解だと思われる選択肢を回答させる．例えば，「画像に写っているイヌ科動物の種類はどれであるか」という問題に対して，「(a)Alaskan Malamute，(b)Siberian Husky，(c)Samoyed，(d)German Shepherd，(e)Gray Wolf，(f)Coyote，(g)Dhole」の選択肢の中から答えるタスクである．

本研究で扱う問題は，次のように定式化される． 多肢選択問題の選択肢集合を C とし，全ワーカの集合を W とする．正答が既知であるテスト問題集合を Q とし，ワーカーは Q の各テスト問題に対して選択肢を一つ選んで回答しているとする．また，正解が $j \in C$ のときに各ワーカー $w \in W$ が各選択肢 $l \in C$ を正解と判定する確率 $\pi^w_{j,l}$ が与えられているとする．選出するワーカー組合せを v 人とし，ワーカ集合 W から $|W|/v$ 個の組合せ $V \subset W$ を選出する．

本問題では，ワーカー集合 W から間違い方の傾向が異なるワーカーを組み合わせることで，多数決の精度が高い複数の組合せを選出し，チーム分けを行うことが目的である．

3.2 単純な多数決

本研究では単純な種類の多数決を考える．まず，単純な多数決について説明する．単純な多数決では選出された各ワーカーの回答の中で最も多く回答された選択肢を正解の選択肢であると推定し，多数決の結果として出力する．最多の回答となった選択肢が複数存在する場合は，それらの選択肢の中から等確率で一つの選択肢を選び，正解の選択肢であると推定する．選出した各ワーカーが選んだ選択肢の組み合わせ $y = y_1, y_2, \cdots, y_k$ に対し，選択肢 $c \in C$ の個数を $S(c)$ とすると，正解として推定される選択肢 $M(y)$ は以下のように表せる．

$$M(y) = \arg\max_{c \in C} S(c)$$

4 期待値の計算

多数決で正解できる確率の期待値を求める方法について説明する．この期待値の計算は後述の計算機実験において組合せの多数決の精度の指標として用いる．選択肢の組み合わせ y が正解する確率 $p(t = M(y) \mid y)$，選択肢の組み合わせが y となる確率 $p(y)$ として，これらから多数決が正解する期待値を計算する．この計算にはベイズの定理を利用し，各ワーカーの混同行列から以下のように求められる．

$$\begin{aligned} E(V) &= \sum_y p(t = M(y) \mid y) p(y) \\ &= \sum_y p(y \mid t = M(y)) p(t = M(y)) \\ &= \sum_y p(t = M(y)) \prod_{i=1}^{k} \pi^{w_i}_{M(y), y_i} \end{aligned}$$

5 愚直な手法

多数決の精度が高いワーカーの組合せを選出するときに考えられる手法は，全通りのワーカーのチーム分けについて，愚直に全ての組合せの期待値を計算し，最も期待値の平均が高かったチーム分けを最良のチーム分けとする方法である．

しかしながら，この愚直な手法には大きな問題点がある．そ

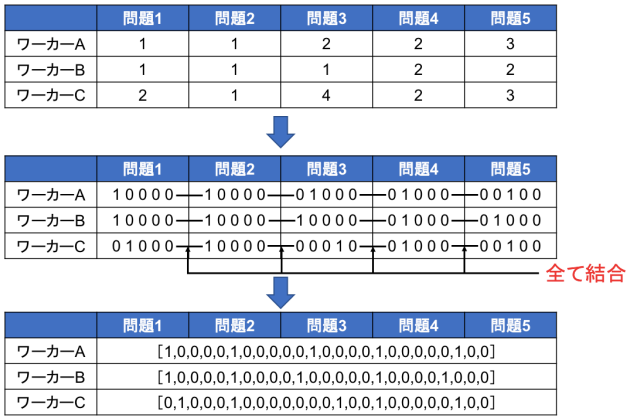

図 1　ワーカーの特徴ベクトル 1 の定義の流れ

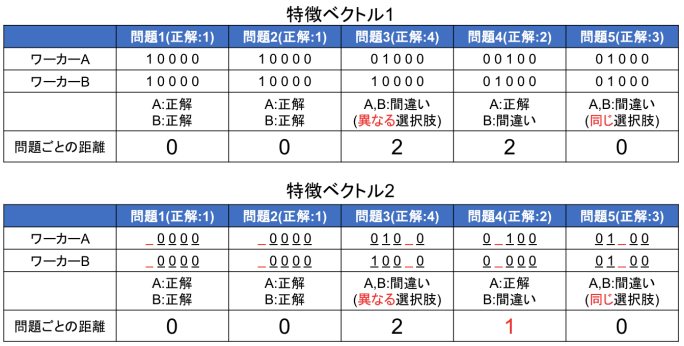

図 2　特徴ベクトル 1 と特徴ベクトル 2 の比較

れは計算量が膨大になってしまうことである．全通りのチーム分けには，$\frac{{}_{|W|}C_v \times {}_{|W|-v}C_v \times \cdots}{(|W|/v)!}$ 通りの分け方が存在する．さらに，期待値の計算には v 人の全ワーカーの組み合わせ，つまり ${}_{|W|}C_v$ 通りの期待値を計算する必要がある．そこで，計算量を小さく抑えつつ，できるだけ多数決の精度の高いワーカーの組み合わせを導出するような手法が必要である．

6　提 案 手 法

提案手法は，ワーカーをクラスタリングし，異なるクラスタから選出したワーカーを組み合わせることで多数決の精度が高くなるようなワーカーの組合せを形成する手法である．多数決が失敗してしまう状況として考えられるのは，一つの間違いの選択肢にワーカーの回答が偏ってしまった結果，間違いの選択肢が多数派となるような場面である．できるだけこのような状況を回避するためには，選出された各ワーカーのそれぞれが間違いの選択肢の一つに偏る確率が低くなるようなワーカーを選出することが重要である．

6.1　ワーカーの特徴ベクトル

提案手法ではまず，後のクラスタリングで利用するために，ワーカーの特徴ベクトルを過去の回答データから定義する．今回は 2 種類のベクトルの定義を説明する．

6.1.1　回答データからの one-hot ベクトル

まずは，テスト問題の回答データからワーカーの特徴ベクトルを定義する方法について説明する．ワーカーの特徴ベクトルを定義する流れについて図 1 に示す．ワーカーの過去の回答データから，各タスクごとに回答の選択肢に対応した one-hot ベクトルへと変換する．ここで変換される one-hot ベクトルとは，次元数が全選択肢の個数と同じであり，回答の選択肢に対応した次元のみが 1，それ以外が 0 となるようなベクトルのことである．例えば，テスト問題の選択肢が 1 から 5 の 5 個あるときに，ワーカーの回答が 2 であったときは (0,1,0,0,0) というベクトルに変換される．この操作を全ての回答データに対して行った後，各回答データの one-hot ベクトルを全て連結し，これをワーカーの特徴ベクトルとして扱う．本論文ではこのように定義したベクトルを特徴ベクトル 1 と呼ぶことにする．また，提案手法では，ワーカー間の距離にはユークリッド距離を用いる．これによって，あるワーカーともう一方のワーカーが同じ問題に対して異なる選択肢で間違ったときにワーカー間の距離が生まれるので，両方が同じクラスタに属しずらくなる．逆に，間違い方が同じであるようなワーカー同士の距離は近くなる．

6.1.2　正解次元を除去したベクトル

6.1.1 におけるベクトルの定義の方法には問題点がある．それは，正解と間違いの選択肢間の距離と異なる不正解の選択肢間の距離が等しくなっているということである．具体的に図 2 を用いて説明する．図 2 のテスト問題 3（正解が 4）においてワーカー A の回答は 2 となっており，ワーカー B の回答は 1 となっている．図 2 ではそれぞれの回答を one-hot ベクトル化しており，それぞれ (0,1,0,0,0) と (1,0,0,0,0) となっている．この 2 つのベクトル間のユークリッド距離は 2 である．テスト問題 4（正解が 2）においてワーカー A の回答は 3 となっており，ワーカー B の回答は 2 となっている．同様にそれぞれの回答を one-hot ベクトル化しており，それぞれ (0,0,1,0,0) と (0,1,0,0,0) となっている．この二つのベクトル間のユークリッド距離は 2 であり，テスト問題 3 における距離と等しくなっている．今回クラスタリングを行う目的は，間違い方の傾向が似ているワーカーを同じクラスタに集めることであるので，間違い方が異なる人の距離が大きくなるようにすることが重要である．そこで，正解の選択肢の次元を one-hot ベクトルから除去することを考える．本論文では，特徴ベクトル 1 から正解の選択肢の次元を除去したベクトルを特徴ベクトル 2 と呼ぶことにする．図 2 において，実際に特徴ベクトル 2 の具体例を示している．特徴ベクトル 2 の各回答の one-hot ベクトルに引かれた赤線が正解の次元であり，その成分を特徴ベクトル 1 から除去している．この結果，問題 4 におけるワーカー A の特徴ベクトル 2 は (0,1,0,0) となり，ワーカー B の特徴ベクトル 2 は (0,0,0,0) となる．問題 4 でのユークリッド距離は 2 から 1 に減少しており，問題 3 の異なる不正解の選択肢間の距離の方が大きくなっている．特徴ベクトル 2 を用いてクラスタリングすると，特徴ベクトル 1 に比べて間違い方の傾向が似ているワーカーがより同じクラスタに集まりやすくなることが期待される．

6.2　k-means 法によるクラスタリング

6.1 の特徴ベクトルからユークリッド距離によるクラスタリ

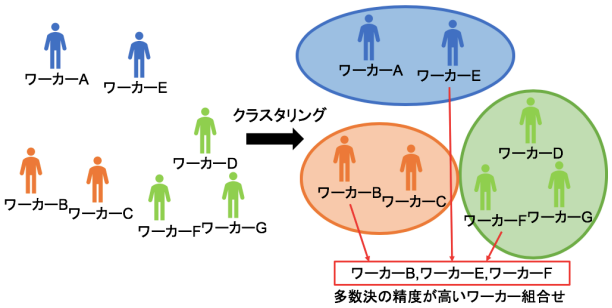

図 3　提案手法のイメージ

ングを行う．提案手法ではクラスタリングには k-means 法を用いた．クラスタ数である k は選出する組合せの人数である v とする．これによって，最後に異なるクラスタからワーカーを選出するときに，各クラスタから一人ずつ選出すれば組合せの人数を v 人とすることができる．

クラスタリングの結果，各クラスタは間違い方の傾向の似通ったワーカーによって構成される．

6.3　異なるクラスタからのワーカー選出

最後にクラスタからのワーカーの選出について説明する．6.2 でのクラスタリングによって各クラスタは間違い方の傾向の似通ったワーカーによって構成されるので，異なるクラスタのワーカーは間違い方の傾向が異なっている．つまり，異なるクラスタのワーカーを選出すると，その組合せの多数決の精度が高くなると考えられる．今回はクラスタ数を組合せの人数である v としているので，各クラスタから一人ずつ選出する．このとき，選出する一人はクラスタから無作為に選ばれる．これによって，v 人の組合せを構成することができる．

今回は全ワーカーを複数の組合せにチーム分けするので，組合せの選出をワーカー全員を選出し終わるまで繰り返す．しかし，k-means 法によるクラスタリングで構成される各クラスタは，属するワーカーの人数にばらつきがあるので，一人ずつ各クラスタから選出していくと，いずれかのクラスタのワーカーが尽きてしまうことに注意する必要がある．なので，ワーカーを選出していく中で，ワーカーを全て選出してしまったクラスタが存在する場合は，その時点でクラスタリングを再度行う．これを繰り返していくことで，間違い方の異なるようなワーカーの組合せを全て選出することができる．

以上をまとめた提案手法のイメージを図 3 に示す．提案手法は計算量は減らしつつできるだけ多数決結果の精度が高くなるような組合せを選出する手法である．

7　評価実験

6 章で述べた提案手法の性能を評価するために行った実験について説明する．

7.1　人工データ

今回は 2 種類の人工データを用いて実験を行なった．全ワーカー数など共通するデータの設定を表 1 にまとめた．今回は，

表 1　データの設定

全ワーカー数 ($\|W\|$)	100
組合せの人数 (v)	5
選択肢数 ($\|C\|$)	5
テスト問題数 ($\|Q\|$)	100

表 2　データ 1 のワーカーの混同行列

		ワーカーの回答				
		1	2	3	4	5
正解	1	0.7	0.075	0.075	0.075	0.075
	2	0.075	0.7	0.075	0.075	0.075
	3	0.075	0.075	0.7	0.075	0.075
	4	0.075	0.075	0.075	0.7	0.075
	5	0.075	0.075	0.075	0.075	0.7

全ワーカー数から組合せの人数を割った 20 個の組合せを作ることになる．一つ目のデータ (データ 1 と呼ぶ) では，全ワーカーの混同行列が共通しており，表 2 に示したような行列になっている．つまり，全ワーカーは正解の選択肢に関わらず正解率が 70%で，それ以外の選択肢を選ぶ確率は一様に 7.5%である．このようなワーカーを人工的に 100 人分用意し，混同行列の確率に基づいて乱数によって回答データを生成した．二つ目のデータ (データ 2 と呼ぶ) では，25 人ごとに混同行列が異なっており，それぞれの行列は表 3 に示している 4 つの行列となっている．データ 2 のワーカーも全員正解率が 70%であるが，回答の傾向が異なっている．一つ目の混同行列では正解が 1 の時に 30%の確率で 2 と答えるが，二つ目の混同行列では 30%の確率で 3 と答え，3 つ目の混同行列では 30%の確率で 4 と答え，4 つ目の混同行列では 30%の確率で 5 と答える．他の選択肢に関しても同様に 25 人ごとに間違い方の傾向が異なる．データ 2 においても，混同行列の確率に基づいて乱数によって回答データを生成した．

7.2　実験環境

実験環境は次の通りである．

- CPU : Intel(R) Core(TM) i5 CPU @ 1.8GHz
- OS : macOS High Sierra 10.13.6
- メモリ : 8GB 1600MHz DDR3
- GPU : Intel HD Graphics 6000 1536 MB

7.3　比較手法

提案手法と比較した手法について説明する．一つ目の比較手法はワーカーを無作為に選び出す手法である．全ワーカーの中から 5 人ずつ無作為に選出し，20 個の組合せを構成する．これを比較手法 1 と呼ぶことにする．二つ目の比較手法は正解率がばらけるように組合せを作る手法である．まず，全ワーカーを回答データにおける正解率で降順にソートする．そして，正解率が 1 位であるワーカーは一つ目の組合せに入れ，正解率が 2 位であるワーカーを二つ目の組合せに入れる．これを繰り返していき，20 位のワーカーを 20 個目の組合せに入れた後，次の 21 位のワーカーは 1 位のワーカーが入っている一つ目の組合せに入れ，22 位のワーカーは二つ目の組合せに入れる．これ

表 3　データ 2 のワーカーの混同行列

		ワーカーの回答				
		1	2	3	4	5
正解	1	0.7	0.3	0	0	0
	2	0	0.7	0.3	0	0
	3	0	0	0.7	0.3	0
	4	0	0	0	0.7	0.3
	5	0.3	0	0	0	0.7

		ワーカーの回答				
		1	2	3	4	5
正解	1	0.7	0	0.3	0	0
	2	0	0.7	0	0.3	0
	3	0	0	0.7	0	0.3
	4	0.3	0	0	0.7	0
	5	0	0.3	0	0	0.7

		ワーカーの回答				
		1	2	3	4	5
正解	1	0.7	0	0	0.3	0
	2	0	0.7	0	0	0.3
	3	0.3	0	0.7	0	0
	4	0	0.3	0	0.7	0
	5	0	0	0.3	0	0.7

		ワーカーの回答				
		1	2	3	4	5
正解	1	0.7	0	0	0	0.3
	2	0.3	0.7	0	0	0
	3	0	0.3	0.7	0	0
	4	0	0	0.3	0.7	0
	5	0	0	0	0.3	0.7

を 100 位のワーカーまで繰り返していく．つまり，正解率上位から 1 人ずつ各組合せに配分していき，20 個の組合せを構成する．これを比較手法 2 と呼ぶことにする．以上の 2 つの手法の性能と提案手法の性能を比較する．

7.4　実 験 結 果

データ 1 とデータ 2 のそれぞれに対して，実験を行なった．提案手法の特徴ベクトル 1，特徴ベクトル 2 のそれぞれを用いて構成された組合せと比較手法による組合せの多数決が正解する期待値は表 4 と 5 のようになった．実験結果のそれぞれの数値は 10 回データを生成し，平均をとったものである．

7.5　実験結果の考察

データ 1 ではいずれの手法においても期待値に大きな差は出なかった．データ 1 におけるワーカーの混同行列では，一様の確率で不正解を選ぶようになっている．提案手法では間違い方の傾向が異なるワーカーを組合せに入れることで，多数決の精度向上を図っているので，データ 1 のような回答データにあまり差が出ていないワーカー集合に対しては効用がなかったことが考えられる．また，データ 1 は全ワーカーの回答の傾向に偏りが起こらないことから，無作為に選出する比較手法でも．比較的回答の傾向が異なる人を組合せに入れることができるので，それぞれの手法の期待値にあまり大きな差が出なかったと考えられる．

表 4　データ 1 での実験結果

手法	期待値	計算時間
提案手法（特徴ベクトル 1）	0.927	0.18s
提案手法（特徴ベクトル 2）	0.927	0.13s
比較手法 1（無作為）	0.9261	0.00015s
比較手法 2（正解率で分配）	0.931	0.00020s

表 5　データ 2 での実験結果

手法	期待値	計算時間
提案手法（特徴ベクトル 1）	0.950	0.12s
提案手法（特徴ベクトル 2）	0.952	0.09s
比較手法 1（無作為）	0.924	0.00015s
比較手法 2（正解率で分配）	0.933	0.00020s

データ 2 では、特徴ベクトル 1 と特徴ベクトル 2 を用いた提案手法がいずれも比較手法より正解する期待値が高い組合せを選出することができた．データ 1 とは違い，データ 2 のワーカーは 25 人ずつ回答の傾向が異なっていることから，組合せを作るときに間違い方の傾向の異なるワーカーを組み合わせることが重要になる．提案手法では，クラスタリングによってうまく回答の傾向が異なるワーカーを選出することができたことにより，比較手法よりも多数決の精度が高い組合せを構成することができた．逆に比較手法の無作為に選出する手法では，組合せに間違い方の傾向が似通ったワーカー同士が入ってしまうことがあるので，期待値が低くなってしまったと考えられる．正解率をばらけさせて組合せを作る手法においても同様の理由で期待値が低くなってしまったと考えられる．

また，特徴ベクトル 2 を用いた提案手法の方が特徴ベクトル 1 を用いた提案手法よりもわずかに期待値の高い組合せを選出することができたが，大きな差は出なかった．特徴ベクトル 2 では，正解と間違いの選択肢間の距離よりも異なる不正解の選択肢間の距離が大きくなるようにしたが，あまり効果が出なかった．今回は間違いが一つの選択肢に偏らないことを重視していたが，一方のワーカーが不正解のときにもう一方のワーカーが正解することで多数決は正解しやすくなるので，特徴ベクトル 2 への変更はあまり効果がなかったと考えられる．

提案手法では，実行可能な計算時間でワーカーの組合せを選出することができた．愚直に全ての組合せの期待値を計算すると計算量が大きくなりすぎるので，提案手法は現実的に利用できる手法であると考えられる．

8　お わ り に

本研究では，全ワーカーの中から多数決の精度が高くなるようなワーカーの組合せを選出することを目的に取り組んだ．ワーカーのテスト問題の回答データから特徴ベクトルからクラ

スタリングを行い，異なるクラスタからワーカーを選出する手法を提案した．異なるクラスタのワーカーは間違い方の傾向が異なっているので，多数決で正解できる確率を上げることができる．性能を評価する実験を行なった結果，回答の傾向が異なるワーカーが存在するようなデータでは，無作為に選出する手法や正解率をばらけた組合せを作る手法よりも多数決の精度が高い組合せを選出することができた．また，組合せを選出する計算時間も現実的な大きさにすることができた．

今後の課題としては，ワーカーの特徴ベクトルの定義の方法やワーカー間の距離関数を改善することが考えられる．まず，本研究ではワーカーの特徴ベクトルはワーカーのテスト問題の回答データから定義したが，クラウドソーソングにおいてはワーカーが必ずしも全てのテスト問題に取り組むとは限らないので，完全な回答データを得られない可能性がある．そこで，ワーカーの混同行列を利用するなど現実的な問題に対応できる手法に改善する必要がある．距離関数については，本研究では単純なユークリッド距離としているが，提案手法の特徴ベクトル 1 と特徴ベクトル 2 で実験結果に差があまり出なかったことからも，正解と間違いの選択肢間の距離や異なる不正解の選択肢間の距離の大きさには改善する必要があると考えられる．また，クラスタリングには k-means 法を利用し，クラスタ数である k を選出する組合せの人数としていたが，ワーカーの回答の傾向によってクラスタ数の最適な値は異なることから，改善する余地がある．k-means 法の k の値を自動で推定する x-means [7] などの手法が研究されており，本研究にも利用することができると考えられる．

9 謝　　辞

本研究は，JST CREST (JPMJCR16E3)，JSPS 科研費 18H03245 の支援を受けたものである．

文　　献

[1] Amazon: Amazon Mechanical Turk, `https://www.mturk.com/`.

[2] Chris, C.-B. and Mark, D.: Creating Speech and Language Data with Amazon ' s Mechanical Turk，Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon ' s Mechanical Turk，pp. 1–12 (2010).

[3] Dawid, A. P. and Skene, A. M.: 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).

[4] Ipeirotis, P. G., Provost, F. and Wang, J.: Quality management on amazon mechanical turk, *Proceedings of the ACM SIGKDD workshop on human computation*, ACM, pp. 64–67 (2010).

[5] Joglekar, M., Garcia-Molina, H. and Parameswaran, A.: Evaluating the crowd with confidence, *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 686–694 (2013).

[6] Joglekar, M., Garcia-Molina, H. and Parameswaran, A.: Comprehensive and reliable crowd assessment algorithms, *2015 IEEE 31st International Conference on Data Engineering*, IEEE, pp. 195–206 (2015).

[7] Pelleg, D. and Moore, A.: X-means: Extending K-means with Efficient Estimation of the Number of Clusters, *In Proceedings of the 17th International Conf. on Machine Learning*, Morgan Kaufmann, pp. 727–734 (2000).

[8] Welinder, P., Branson, S., Perona, P. and Belongie, S. J.: The multidimensional wisdom of crowds, *Advances in neural information processing systems*, pp. 2424–2432 (2010).

[9] Whitehill, J., fan Wu, T., Bergsma, J., Movellan, J. R. and Ruvolo, P. L.: Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise, *Advances in Neural Information Processing Systems 22* (Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I. and Culotta, A., eds.), Curran Associates, Inc., pp. 2035–2043 (2009).

[10] Wu, T., Chen, L., Hui, P., Zhang, C. J. and Li, W.: Hear the whole story: Towards the diversity of opinion in crowdsourcing markets, *Proceedings of the VLDB Endowment*, Vol. 8, No. 5, pp. 485–496 (2015).