

クラウドソーシングによるリバースを題材にした意見統合手法の検討

佐々木 優[†] 馬場 雪乃^{††} 鹿島 久嗣^{††} 森嶋 厚行^{†††}

[†] 筑波大学大学院図書館情報メディア研究科 〒305-8550 茨城県つくば市春日 1-2

^{†††} 筑波大学知的コミュニティ基盤研究センター/図書館情報メディア系 〒305-8550 茨城県つくば市春日 1-2

^{††} 京都大学大学院情報学研究科 〒606-8501 京都市左京区吉田本町 36-1

E-mail: [†]yu.sasaki.2015b@mlab.info, ^{††}{baba,kashima}@i.kyoto-u.ac.jp, ^{†††}mori@slis.tsukuba.ac.jp

あらまし インターネットの出現により、多人数による意見の集約手法として、単純な多数決以外の様々な方法が可能になり、インターネット上の群衆による意思決定方法は近年注目されている。本稿では、意見集約の問題の題材として、ルールが広く知られているリバースを用いて、マイクロタスク型のクラウドソーシングを行い、実際にいくつかの手法を比較した結果を報告する。

キーワード クラウドソーシング, データ統合, リバース

1. はじめに

インターネットの出現により、多人数による意見集約に関して、単純な投票による多数決以外の様々な方法が可能になった。そのような、インターネットを用いた多人数による意見集約プロセスの研究は近年注目を集めている。

様々な意見の収集・統合の中でも、「簡単でない問題に対して解を求める」ための意見収集・統合手法は広い応用を持つと考えられる。これは、画像に猫が写っているか否かといった比較的簡単な問題とは異なる。例えば、ネット上で流れているニュースの真偽を明らかにするといった問題は、簡単ではない。画像に猫が写っているか否かという問題であれば、単純な多数決をとる方法が有効である。しかし、ニュースの真偽を明らかにするといった問題の場合は、必ずしも多数決が有効であるとは言いがたい。真偽を求める方法としては、ニュースの発信元などを頼ってニュースの情報源を問い合わせるという情報収集の方法や、関連情報を照らし合わせて情報統合する方法などが考えられる。このように、解を求めることが簡単でない問題に対しては単純な多数決などの手法以外のアプローチも有効と考えられる。

本論文では、リバースを題材とし、クラウドソーシングでより良い解を求めるための意見収集・統合手法について議論する。リバースを題材とした理由は、ルールが単純で、これまでリバースをやったことがない人でもルールを理解しやすく、評価がしやすいためである。

本稿は、著者らによる以前の発表 [6] の枠組みをベースに、比較のための要素として回答の統合手法の違いを追加し、その上で、不特定多数の参加者による実験を行った結果を報告するものである。

本論文は次のように構成される。2. 節では、関連研究について述べる。3. 節では、設計した方法を形式化し、設計する際に用意した比較項目について述べ、4. 節では、3. 節の比較項目を組み合わせたタスク設計の例について述べる。5. 節では、実験とその評価について述べる。6. 節では、まとめと今後の課題を

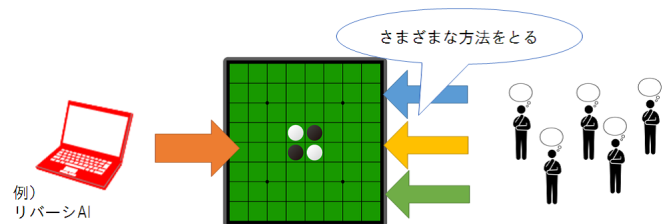


図1 不特定多数の人々によって次の一手を決定するより良い方法は？

述べる。

2. 関連研究

1990年代、当時のチェス世界王者とクラウドソーシングによる集合知「ワールド」の対戦が行われた。「ワールド」とは、世界中のチェス愛好家のべ5万人で構成されたチームで、彼らの次の一手は多数決によって決められた。この対戦は2回行われ、いずれも「ワールド」は敗退した。しかし、2回の対戦での「ワールド」の次の一手の決定方法は、それぞれ条件が異なり、この違いは戦局に違いを与えたと言われている [9]。最近では、人と計算機がチームを組んで行う Advanced Chess の分野 [10] で、終盤に強くミスもしない計算機の強みと、形勢判断能力に優れた人間の強みを生かせば、人単独や、計算機単独よりも強いという結果が示されている。本研究は、リバースというシンプルなゲームを対象に、多様なタスクデザインを試す事によって、どの要素が意見統合にどう影響を及ぼすのかを詳細に明らかにする事を目標とする。

簡単でない問題に対して解を求めるときにクラウドソーシングを用いる研究は、これまでも行われてきている。Lotoshら [7] は、旅行計画の作成を木の探索としてモデル化し、群衆はノード全てに対して投票して点数をつけ、探索を行っている。本研究でも、リバースの先読みを木の探索とすることができるが、多数決を行って複数あるノードの中から1つのノードを選ぶなど、枝刈りを行いながら探索している点で異なる。また、米良ら [8] は、解となる仮説の入手とその検証を、1つのタスク

表 1 タスク設計で着目した要素

要素	選択肢
先読み方法	各自, 分担
ワーカの入力	置くべき場所, 置くべきと置くべきでない場所
ワーカに見せるもの	盤面のみ, 置ける場所
回答の統合方法	単純多数決, 強さ申告, 各手の自信度

で同時に実行しているが、本研究では、解探索を明示的に群衆に依頼するような複数種類のタスクデザインが研究対象である。

また、小山ら [1] は、ラベル統合の問題に関して、ワーカの回答の自信度を用いて解となるラベルを推定する確率的手法を提案している。本研究でも回答の統合手法の 1 つに自信度を用いているが、この手法では単純にもっとも自信度が高い回答を解として採用している。

3. 定式化と比較項目

リバースの盤面 $Board$ と次の手の色 $color$ が与えられた時、次に打つ可能性がある手の候補の集合を $Moves(Board, color) = \{m_1, \dots, m_n\}$ とする。このとき、この問題は、 $m_x \in Moves(Board, color)$ なる m_x を一つ決定する関数 $Next$ として定式化できる。

$$m_x = Next(Board, color)$$

タスク設計は様々な要素に着目して設計することが考えられるが、今回は表 1 の要素に着目してタスクを設計する。

(1) どう先読みするか：次の手を打つための先読みをどう行うかに着目する。各人が個々に先読みをする方法を、各自先読み、複数人で分担する方法を分担先読みと呼ぶ。各自先読みでは、ワーカが頭の中で何手先かまでを先読みするが、分担先読みでは各自先読みでワーカが行っている先読みを明示的にタスクとして用意する。このタスクを分割し、複数人で分担して先読みを行う。先読みする幅 k 、深さ l をパラメータとし、 $WkDl$ と表す。

(2) なにをワーカが入力するか：次に打つ手を選ぶ時、どんな手をワーカが入力するかに着目する。打つべき手を入力する方法をポジティブのみ (P)、打つべき手と打つべきではない手を入力する方法をポジティブネガティブ (PN) と呼ぶ。

(3) なにをワーカに見せるか：次に打つ手を選ぶ時、ワーカに見せる情報に着目する。 $Board$ のみ見せる (ϕ)、入力候補である $Moves(Board, color)$ を見せる (候補あり) がある。

(4) どのように回答を統合するか：単純多数決による方法、自己申告による自信度を利用する方法がある。リバースにおいて自信度は様々なものが考えられるが、今回は、自己申告によるリバースの強さを利用する方法 (強さ申告)、各手毎の自己申告による自信度を利用する方法 (各手の自信度) を用いる。

今回、回答の統合方法は単純多数決を用いることが多かったため、単純多数決の場合は方法名のうち統合方法の要素の表記を省略している。

表 2 リバースの強さの自己申告

5	日本オセロ連盟の段位を持っている
4	日本オセロ連盟の級位を持っている
3	段や級は持っていないが、リバースは強い方だ
2	勝つテクニックをいくつか知っているが、強いとは言えない
1	勝つテクニックを全く知らず、弱い

表 3 各手の自信度

5	とても自信がある
4	まあまあ自信がある
3	どちらでもない
2	あまり自信が無い
1	全然自信がない

	a	b	c	d	e	f	g	h
1								
2								
3				●	○			
4				●	○			
5				●	○			
6								
7								
8								

黒：3白：3

黒の手番です



図 2 各自先読み-P-候補ありのタスク画面 (出典：佐々木 (2016) [6])

4. タスク設計の例

次に、3. 節の要素を組合せたいくつかのタスクデザインを説明する。

4.1 各自先読み-P

各自先読み-P は、各自がそれぞれ先読みを行い、次に置くべきマスを投稿する方法である。回答されたマスの中で最も投票数が多かったマスを次の一手に決定する。最も投票数が多かったマスが複数存在した場合は、それらのマスの中からランダムで決定する。

形式的には次のように定義される。 $M = Moves(Board, color)$ としたとき、各候補 $m_i \in M$ が得た投票数 $positive(m_i)$ の大小で決まる M 上の順序関係を \leq_p とする。さらに、 M 中の全ての要素を、順序関係 \leq_p で降順 (同点の場合にはランダム) に並べた列を返す関数を $Sort_{\leq_p}(M)$ 、列 S の先頭から 1 番目の m_k を取り出す関数を $S[1]$ とする。この時、次に置くマスを決定する式は次のようになる。

$$Next(Board, color) \equiv (Sort_{\leq_p}(M))[1]$$

各自先読み-P-候補ありのタスク画面を図 2 に示す。図 2 のように、ワーカは次に石を置くべきだと思うマスを、あらかじめ提示された置けるマスの選択肢の中から回答する。

各自先読み-P-候補無しでは、置けるマスの選択肢を提示せず $Board$ のみを表示し、ワーカが自由に座標を入力する。し

	a	b	c	d	e	f	g	h
1								
2								
3								
4			○		●			
5			○	○	○	●		
6						●		
7						●		
8								

黒：4白：4

黒の手番です

打つべき場所を選んでください

○ c4 ● b5 ○ c6 ● e6 ○ c3

打つべきでない場所を選んでください

○ c4 ● b5 ○ c6 ● e6 ○ c3

送信

候補A

	a	b	c	d	e	f	g	h
1								
2								
3								
4			●		●			
5			●	○				
6								
7								
8								

候補B

	a	b	c	d	e	f	g	h
1								
2								
3								
4			●	●	●			
5				●	○			
6								
7								
8								

この手番で黒が勝てる盤上はどれですか？

A
B

図3 各自先読み-PN-候補あり タスク画面 (出典: 佐々木 (2016) [6])

たがって、ワーカにとって気がつかない候補が存在する場合があります。

4.2 各自先読み-PN

各自先読み-PNは、置くべきマスと置くべきでないマスの回答を求める方法である。

m_i の「置くべき」の投票数を $positive(m_i)$, 「置くべきでない」の投票数を $negative(m_i)$ とする。各候補 $m_i \in M$ の $positive(m_i) - negative(m_i)$ の大小で決まる M 上の順序関係を \leq_{pn} とする。さらに、 M 中の全ての要素を、順序関係 \leq_{pn} で降順 (同点の場合にはランダム) に並べた列を返す関数を $Sort_{\leq_{pn}}(M)$ とする。この時、次に置くマスを決定する式は次のようになる。

$$Next(Board, color) \equiv (Sort_{\leq_{pn}}(M))[1]$$

各自先読み-PN-候補ありのタスク画面を図3に示す。図3のように、ワーカは次に自分の色の石を「置くべき」だと思うマスと「置くべきではない」と思うマスを選択し回答する。複数のマスに同じ「置くべき」の票数が入っていた場合、各自先読み-Pではそれらからランダムで決定したが、PNでは「置くべきでない」の投票数を考慮する事により、より良いマスを選択させる事を狙う。

4.3 各自先読み-P-候補あり-自信度

これは、回答の中でも、ワーカが自己申告した自信度が高い回答を採用する方法である。具体的には、各自先読み-P-候補ありに自信度を回答する項目を加えた方法である。今回、この自信度は表2~表3に示した5段階の中から選択する。

次の一手を決める流れを説明する。まずワーカは次に石を置くべきだと思うマスを、あらかじめ提示された置けるマスの選択肢の中から選び、同時に自信度も回答する。回答の中からもっとも自信度が高い人が選んだマスを次の一手に決定する。自信度をもっとも高い人の回答が複数あった場合には、その中からもっとも投票数が多かったマスを次の一手に決定する。このとき、もっとも投票数が多かったマスが複数存在した場合は、

図5 分担先読み 勝敗予想タスク (佐々木 (2016) [6] の図を一部修正)

それらのマスの中からランダムで決定する。

4.4 分担先読み

分担先読みは、各自で先読みをして置くべきマスを選ぶのではなく、分担して先読みを行う事によって、次の手を決定する方法である。図4に、最も単純な場合の分担先読みのタスクの流れを示す。分担先読みでは二種類のタスクを用いる。

多数決タスク 石を置くべきマスを選択する。タスク画面は各自先読みのタスクと同じである。ただし、各自先読みの場合には、黒の担当の人は、黒の手を入力するだけであったが、分担先読みの場合は、自チームの立場だけではなく、相手の立場になった手も打つ事になる。

勝敗予想タスク 多数決タスクにおいて投票されたマスのうち、投票数が多かった上位 N 個それぞれに、仮に石を置いた場合の盤面をタスク画面に表示する。その盤面をみて今後自分の石がどうなるか、「勝つ」「わからない」「負ける」のどれかを予想する。タスク画面を図5に示す。

例として、先読み数2の場合のタスクの流れを説明する。まず、システムは黒の手を入力する多数決タスクを生成する。黒の手が入力されると、その結果を見て多数決で上位2件のものに対して、今度は白の立場になって行う多数決タスクを生成する。その後、多数決で上位2件について勝敗予想タスクを生成し、その結果から最終的な手を決定する。

各候補 $m_i \in M$ が勝敗予想タスクにおいて勝つと投票された数 $win(m_i)$ の大小で決まる M 上の順序関係を \leq_{win} と表記すると、分担先読み **WkD1-P** は次のように定義される。

$$Next(Board, color) \equiv \max_{\leq_{win}}(M')$$

$$\text{where } M' = (Sort_{\leq_p}(M))[1, 2, \dots, k]$$

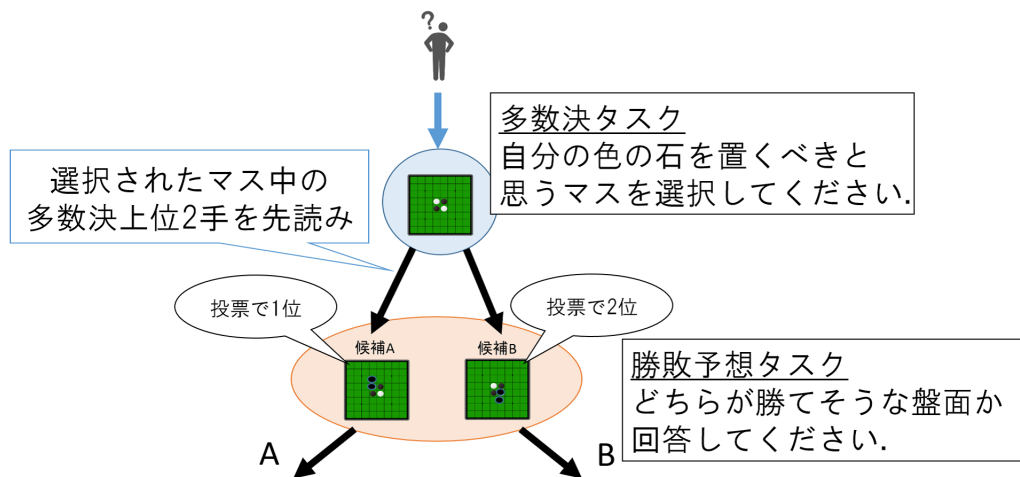


図4 分担先読み W2D1 の流れ (佐々木 (2016) [6] の図を一部修正)

5. 実験

4. 節で説明した中で、各自先読み-P-候補あり、各自先読み-PN-候補あり、各自先読み-P-候補あり-各手の自信度、各自先読み-P-候補あり-強さ申告、分担先読み W2D1-P-候補ありのタスク設計を用いてリバースの試合を行った。実験設定とその評価について説明する。今回の実験で行った方法では、全て候補ありの要素を用いたため、候補ありの表記は省略した。また、分担先読みは W2D1 しか行っていないため、W2D1 の表記は省略した。

5.1 設定

タスクは Crowd4U [2] 上で実装した。実験参加者は Yahoo! クラウドソーシング [4] 上で集めた。実験参加者は実験前に、自分のリバースの強さを表 2 に示した 5 段階で自己申告した。実験参加者の強さ申告を表 4 に示す。

実験参加者は 1 試合に 1 回のみ次に置くべきマスの投票ができる。実験参加者チーム (以下、群衆) の次の一手は、手番ごとに 5 票投票が入った時点で多数決を行って決定する。リバースは、1 試合におよそ 30 手必要であるため、各自先読みのタスク設計を用いた試合に必要な票数 (参加者数) はおよそ 150 票 (人) である。また、今回分担先読みのタスク設計を用いた試合に必要な票数は、多数決タスクに 5 票、勝敗予想タスクで 10 票得て次の一手を決定することとしたため、(5 票 + 10 票) × およそ 30 手番でおよそ 450 票である。ただし、置けるマスがない場合や置けるマスが 1 つしかない場合は投票を必要としなかったり、試合の進行によってはパスなどの理由により 30 手分より多く石を打ったり、少なく石を打ったりするため、実際の投票数は上下する。各方法の実験参加者数を表 6 に示す。

群衆の対戦相手には、オセロ世界大会出場者である中島哲也オセロ 8 段が開発したリバース AI [3] を使用した。このリバース AI は 3 段階の強さのうち最も易しい設定にした。

今回は、群衆は先手、リバース AI は後手を担当した。

試合の勝敗と石数について、表 5 に示す。

表 4 各方法の実験参加者の強さ申告の割合

強さ	各自先読み-P	各自先読み-PN	各自先読み-P-各手の自信度	各自先読み-P-強さ申告	分担先読み-P
5	0%	0%	0%	1%	0%
4	0%	0%	0%	0%	1%
3	16%	21%	19%	17%	14%
2	52%	42%	39%	41%	37%
1	32%	37%	42%	42%	48%

(※小数点以下四捨五入)

表 5 石数と勝敗

方法	群衆	リバース AI	勝者
各自先読み-P	19	45	リバース AI
各自先読み-PN	13	51	リバース AI
各自先読み-P-各手の自信度	29	35	リバース AI
各自先読み-P-強さ申告	30	34	リバース AI
分担先読み-P	21	43	リバース AI

表 6 参加者数

方法	参加者数
各自先読み-P	125
各自先読み-PN	125
各自先読み-P-各手の自信度	145
各自先読み-P-強さ申告	145
分担先読み-P	345

5.2 評価方法

評価では、WZebra [5] というソフトを使用する。それぞれの試合の棋譜を WZebra に入力し、WZebra の評価関数が出力した評価値を評価に使用する。この評価の際、完全読みでは時間がかかるため、読みの深さは終盤までは 24 手先読みし、残り 26 マスとなった時点から完全先読みしたものとした。

5.3 結果

各方法のタスク設計で行った実験の評価値のグラフを図 6 ~ 図 8、評価値の平均を表 7 に示す。途中で先読みの結果勝敗が決まった時やパスの時は、評価値は出力されないため、このときの値の点はグラフ中にプロットしていない。評価値は、盤面を評価して算出される。基本的にどの方法も、負の値をとることが多かったが、各自先読み-P-各手の自信度のみが正の値を

とることがあった。

また、各方法の票の散らばりを比較するために、各自先読み-P、各自先読み-PNの各手のジニ係数の平均を計算した(表8)。ジニ係数は0から1の間で値を取り、値が1に近い程より票がまとまっている事を示す。予備実験時[6]に比べて、今回の実験の方が票がまとまっていたことがわかる。

さらに、ワーカの強さ申告毎の評価を比較するために、各方法のワーカの強さ申告の組み合わせとその回答結果の盤面とその前の盤面の評価値の差の平均の関係を図9~図12に示す。これらの図の縦軸は前の評価値と差の平均、横軸はワーカが申告した強さである。横軸は、強さが高い人が多い組み合わせ順で並べた。各自先読み-P、PNは、多数決を行い、複数の回答を参考にして1つのマスを選びだしたため、その手番に参加したワーカの強さ申告の組み合わせを横軸とした。各自先読み-P-各手の自信度、強さ申告は、ワーカが回答した各手の自信度や、強さ申告をもとに複数の回答の中から1つの回答だけを採用したため、採用された回答の強さ申告を横軸とした。また、各自先読み-P-各手の自信度について、ある手番でもっとも各手の自信度が高い回答が複数あった場合、それら回答のリバーシの強さを平均して四捨五入した値をその手番で選ばれた回答で申告された強さとした。分担先読みは1つの手番に必要としたワーカ数が異なるため、ここでは比較していない。

5.4 考察

今回は限られた実験ではあるが、今回の範囲では次のことが示唆される。

- 表7からは、各自のリバーシの強さの申告よりも、各手ごとの自信度のほうが、より良い手を求める手掛かりとなる。
- 同表から、分担先読みもより良い手を求めるのにある程度の効果がある。
- 表8から、オフラインで依頼した人々による手と比べて、実際にクラウドソーシングでワーカをリクルートしたほうが、手がまとまる傾向にある。ただし、この主な要因が、クラウドソーシングによってリクルートしたためかは、今回の実験からは明らかではない。オセロの有名な戦略の一つに、相手の打てる箇所を減らし、自分の打てる箇所を増やすように打つというのが、調査したところ、オフラインで依頼した人々による試合では、自分が打てる箇所の試合での合計が281(各自先読みP)と268(各自先読みPN)、クラウドソーシングによってリクルートした今回の実験では145(各自先読みP)と169(各自先読みPN)であった。ジニ係数は、選択肢の数が減ると値が小さくなる傾向になるため、その影響の可能性はある。この点については今後さらなる検証が必要である。

6. まとめと今後の課題

本稿では、リバーシを題材として、難しい問題に対するインターネット上の群衆による意見統合手法のよりよい方法について知見を得ることを目標に実験を行った。今後は、今回得た結果から示唆された知見のより厳密な検証と、その知見を活用したより良い意見統合手法を明らかにしたい。

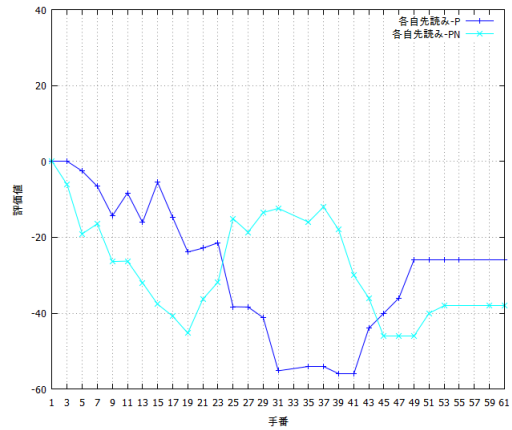


図6 各自先読み-P,PNの評価値

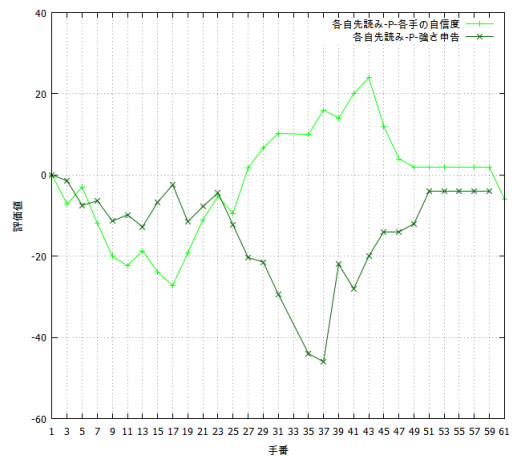


図7 各自先読み-P-各手の自信度、強さ申告の評価値

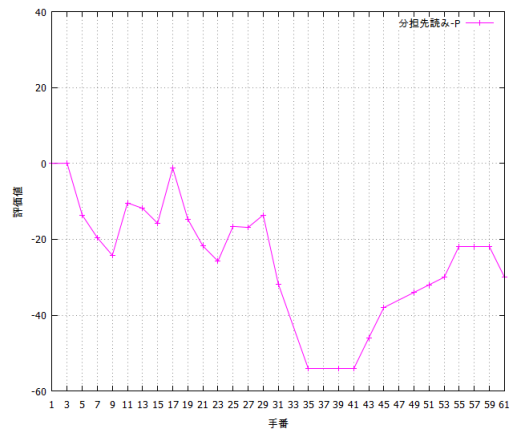


図8 分担先読み-Pの評価値

謝辞

本研究に協力していただいている中島哲也オセロ8段に感謝します。本研究の一部は、JST CREST および JSPS 科研費(#25240012)の支援による。

文献

[1] Satoshi Oyama, Yukino Baba, Yuko Sakurai, Hisashi

表 7 評価値の平均

方法	評価値の平均
各自先読み-P	-27.84
各自先読み-PN	-27.92
各自先読み-P-各手の自信度	-1.94
各自先読み-P-強さ申告	-24.13
分担先読み-P	-13.29

(※小数点第三位以下四捨五入)

表 8 ジニ係数の平均

方法	予備実験	今回の実験
各自先読み-P	0.356	0.706
各自先読み-PN	0.282	0.720

(※小数点第四位以下四捨五入)

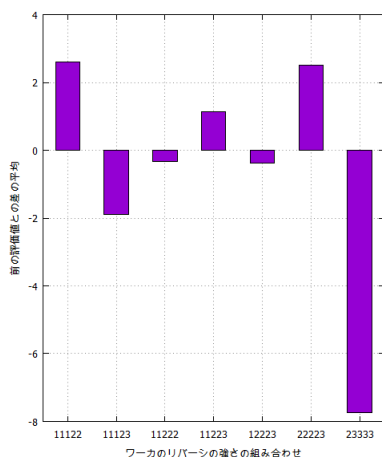


図 9 “各自先読み-P”における“ワーカの強さ申告の組み合わせ”と“その回答結果の盤面とその前の盤面の評価値の差の平均”の関係

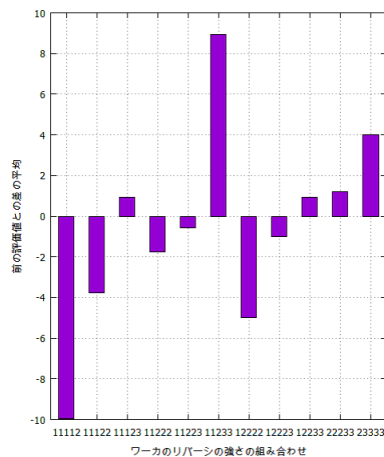


図 10 “各自先読み-PN”における“ワーカの強さ申告の組み合わせ”と“その回答結果の盤面とその前の盤面の評価値の差の平均”の関係

Kashima. “Accurate Integration of Crowdsourced Labels Using Workers’ Self-reported Confidence Scores”. 23rd International Joint Conference on Artificial Intelligence (IJCAI), 2013

[2] “Crowd4U - クラウドフォーユー: 非営利・公益・学術目的のクラウドソーシング・マイクロボランティアプラットフォーム -

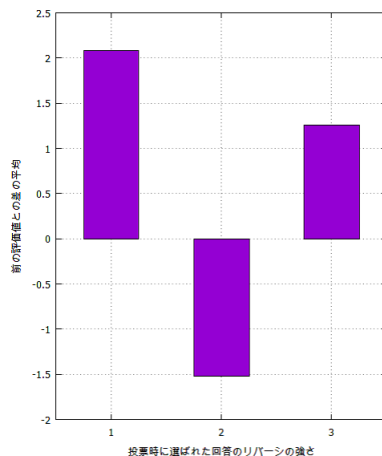


図 11 “各自先読み-P-各手の自信度”における“回答統合時に採用された回答を行ったワーカの強さ申告の値”と“その回答結果の盤面とその前の盤面の評価値の差の平均”の関係

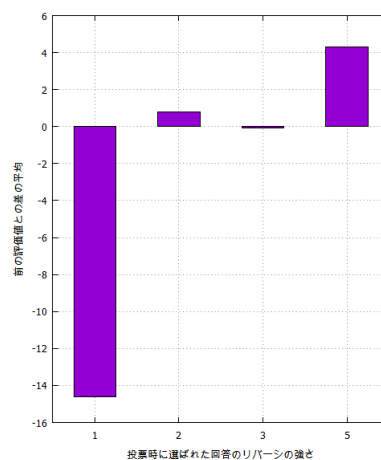


図 12 “各自先読み-P-強さ申告”における“回答統合時に採用された回答を行ったワーカの強さ申告の値”と“その回答結果の盤面とその前の盤面の評価値の差の平均”の関係

Crowd4U”. Crowd4U. <http://crowd4u.org/>, (2017-01-13).

[3] “オセロ・無料ゲーム - Othello! JAPAN”. Othello! JAPAN . <http://www.othello.org/play/>, (2017-01-13).

[4] “クラウドソーシングで稼ごう - Yahoo!クラウドソーシング”. Yahoo!クラウドソーシング. <http://crowdsourcing.yahoo.co.jp/>, (2017-01-13).

[5] “Gunnar’s Othello page”. <http://radagast.se/othello/>, (2017-01-13).

[6] 佐々木 優, 平木 理恵, 馬場 雪乃, 森嶋 厚行, 鹿島 久嗣. “マイクロタスク型クラウドソーシングによるリバースの試み”. 2016 年度人工知能学会全国大会 (第 30 回). 2016.

[7] Lotosh Ilia, Tova Milo, and Slava Novgorodov. “CrowdPlanr: Planning made easy with crowd.” Data Engineering (ICDE), 2013 IEEE 29th International Conference on. IEEE, 2013.

[8] 米良俊輝, 平木理恵, 若宮翔子, 森嶋厚行, 荒牧英治. “クラウドソーシングを用いた仮説入手・検証の自動化”. DEIM Forum. 2016.

[9] マイケル・ニールセン. オープンサイエンス革命. 紀伊國屋書店, 2013.

[10] Rasskin-Gutman, Diego. Chess metaphors: artificial intelligence and the human mind. MIT Press, 2009.