3M1-OS-07c-1

クラウドソーシングにおける必要ワーカ数の動的決定方法の提案

A dynamic method for determining the optimal number of workers in crowdsourcing

櫻井 祐子[*1] 岡 雅晃[*2] 沖本 天太[*3] 篠田 正人[*4] 横尾 真[*2]
Yuko Sakurai Masaaki Oka Tenda Okimoto Masato Shinoda Makoto Yokoo

[*1]九州大学，JST さきがけ Kyushu University, JST PRESTO
[*2]九州大学 Kyushu University
[*3]新領域融合研究センター Transdisciplinary Research Integration Center
[*4]奈良女子大学 Nara Womens' University

We propose a method for dynamically determining the optimal number of workers in crowdsourcing. Crowdsourcing services provide excellent environments for aggregating data/labels from many more workers with much cheaper cost. In general, a requester determines the number of workers in advance. However, according to the abilities of aggregated workers, it's desirable to dynamically decide the number of workers to achieve the quality of a task result that a requester wants to guarantee. We formalize this problem as the optimal stopping problem and propose a new method by utilizing a worker's confidence.

1. はじめに

近年，ネットワーク環境の普及に伴い，インターネット上では様々なサービスが提供されている．特に，ここ数年で成長が著しいサービスの一つとして，Amazon Mechanical Turk に代表されるクラウドソーシングがある．クラウドソーシングは「群衆の知」の概念が根本にあり，多数の人々の知恵や力を利用することによって問題を解決するサービスである．クラウドソーシングのサービスを利用することで，世界中の人々に安価にタスクを実行させることが可能となっている．さらに，現在の計算機の能力では解決が難しいが，人間では容易に解決可能な問題などに対して，人間をプログラムの一部として機能させることで解決を目指すアイデアに基づく Human Computation など，今後の計算機科学における計算機と人間の融合に関する有望な研究領域になっている [鹿島 12, John 10, Law 11].

本論文では，クラウドソーシングにおいて，ワーカ数を事前に定めるのではなく，ワーカの解答状況に応じて，ワーカ数を動的に決定する手法の提案を行う．クラウドソーシングでは，リクエスタが設定した時間内で，ワーカらがそれぞれ異なる時刻にタスクを実行する．タスクが複数選択肢から正解を選択するラベル選択問題では，一般に多数決によってリクエスタは正解を判定する．このとき，リクエスタは少ない費用で効率的に高い正解判定率を得たいと考えるであろう．この目的に対して，事前に決定したワーカ数に至らずとも，同程度の正解判定率を少ないワーカ数で得られる，もしくは正解判定率の改善が見込めないのであれば，途中で打ち切ることが望ましい．

そこで，我々は，本問題を最適停止問題 (Optimal stopping problem) [Ferguson 89, 穴井 2000] として定式化し，リクエスタの期待利得が最大化するワーカ数を動的に決定する手法の提案を行う．また、文献 [Sakurai 13] ではワーカの能力は必ずしも同一でないことに着目し，解答ともに自信を正直に申告させ, ワーカを自信の有無によって 2 つに分類することでの重み付き投票に基づく多数決によって正解判定率を向上させる手法が提案されているが, この手法においてもリクエスタの期待利得を最大化させるようなワーカ数を動的に決定することが可能であることを示す.

連絡先: 櫻井 祐子，九州大学大学院システム情報科学研究院，福岡市西区元岡 744，ysakurai@inf.kyushu-u.ac.jp

2. ワーカ数最適化問題

本章では，ワーカの能力は区別できないものとし，リクエスタは単純多数決で正解判定を行うときの最適なワーカー数を動的に決定する手法を提案する．

まず，本論文では，yes/no の判断問題や画像の二値分類など，解候補が 2 つしかない二択問題を対象とする．複数のワーカの解答によってリクエスタが $l \in \{1,2\}$（binary label）の判定を行う．ワーカは最大で n 人であるとし，n は奇数とする．リクエスタは l に関する事前情報を持っていないとする．

ワーカは逐次的にタスクを実行し，各時刻 k $(1 \leq k \leq n)$ にタスクを実行するワーカをワーカ k とする．ネットワーク環境で各々異なる場所からタスクを実行するため，ワーカの解答は互いに独立である．すなわち，各ワーカは独立に確率 p で正しい解答をするとする．ただし，$p \in [0.5, 1]$ は密度関数 f に従う未知のパラメータ（$\int_{0.5}^{1} f(p)dp = 1$）であることを仮定する．

ワーカ n 人の解答ベクトルを $\vec{a} = (a_1, a_2, \cdots, a_n) \in \{0,1,2\}$ とし，$a_i = 0$ のときは i 番目のワーカにはまだタスクを実行させていないことを意味する．この $\vec{a}$ に対し，$\vec{a}_k = (a_1, \cdots, a_k, 0, 0, \cdots, 0)$ とし，k 番目のワーカまでタスクを実行させた場合に得られている解答ベクトルとする．$\vec{a}_k$ の要素のうち 1 が α 個，2 が β 個 $(\alpha+\beta=k)$ であるときの実現確率 $P(\vec{a}_k)$ は

$$
\begin{aligned}
P(\vec{a}_k) &= \frac{1}{2}\int_{0.5}^{1} p^{\alpha}(1-p)^{\beta} f(p)dp + \frac{1}{2}\int_{0.5}^{1} (1-p)^{\alpha} p^{\beta} f(p)dp \\
&= \frac{1}{2}\int_{0}^{1} p^{\alpha}(1-p)^{\beta} f(p)dp
\end{aligned}
$$

である．ここで，$p \in [0, 0.5]$ に対し $f(p) = f(1-p)$ と拡張している．

次に，停止時刻 (打ち切りルール) τ を定義する．τ は解答ベクトル $\vec{a} = (a_1, a_2, \ldots, a_n)$ から時刻 $\{1, 2, \ldots, n\}$ への関数

であり,「いつワーカを打ち切るか」を表し, $\tau \leq k$ かどうかは $\vec{a}_k$ で定まるものである．本論文では，ワーカの期待利得によって停止時刻を決定する．リクエスタは，ワーカらから集めた解答ラベルによって最終的な解答を決定するが，その解答が正解である場合には r_C のグロス効用，不正解の場合には r_I のグロス効用が得られるものとする．さらに，各ワーカに払う報酬はそれぞれ c とする．停止時刻 τ で打ち切ったときの正解確率を x_τ とすると，リクエスタの利得は,

$$x_\tau r_C + (1 - x_\tau) r_I - \tau c \tag{1}$$

となる．

従って，リクエスタにとって，最適な停止時刻とは式 1 で定義した利得の期待値を最大化するものである．すなわち，解答ベクトル $\vec{a}_k$ が得られたとき，ここで停止するかどうかは,

$$\arg\max_{\tau} E(x_\tau r_C + (1 - x_\tau) r_I - \tau c) \tag{2}$$

によって定まる τ において $\tau(\vec{a}_k) = k$ となるかどうかによって決定することとなる．リクエスタの期待利得最大化問題として本問題を定式化することで，打ち切りに関する様々な条件を包含して表現することができる．

以下、期待利得を最大化する τ を決定するために, 途中段階での正解判定確率を以下のように求める．$k = \alpha + \beta$ 人目で $l = 1$ に α 票，$l = 2$ に β 票入っているとき，$l = 1$ と判定した結果が正しい確率は

$$\left\{ \int_{0.5}^{1} p^\alpha (1-p)^\beta f(p) dp \right\} \Big/ \left\{ \int_{0}^{1} p^\alpha (1-p)^\beta f(p) dp \right\}$$

と表せる．この値を $g(\alpha, \beta)$ とする．$\alpha > \beta$ のときは $l = 1$, $\alpha < \beta$ のときは $l = 2$ の方が正しい確率の高い判定となり，正解判定確率 $x_{\alpha,\beta}$ は

$$x_{\alpha,\beta} = \max\{g(\alpha, \beta), g(\beta, \alpha)\}$$

となる．

そして，次に $k+1$ 人目を採用するとき, その $k+1$ 人目の解答 a_{k+1} の確率分布は

$$P(a_{k+1}|\vec{a}_k) = \frac{P(a_1, a_2, \ldots, a_k, a_{k+1}, 0, \ldots, 0)}{P(a_1, a_2, \ldots, a_k, 0, \ldots, 0)} = \frac{P(\vec{a}_{k+1})}{P(\vec{a}_k)}$$

という条件付き確率で表される．これは，k 人目までのワーカの解答に基づき，事前情報を修正して得る予測値である．最初に定義したように，$l = 1$ と $l = 2$ という正解の可能性は同等としており，リクエスタにとって最初のワーカの解答が $1, 2$ どちらになるかは五分五分であったが，その後の解答がどちらになるかは「それまでの解答結果」によって偏りを持って予測することとなる．従って，k 人目で $l = 1$ に α 票，$l = 2$ に β 票入っているとき，次のワーカの解答を聞いてから止めるとすれば，その正解判定確率の期待値は

$$P(a_{k+1} = 1|\vec{a}_k) x_{\alpha+1,\beta} + P(a_{k+1} = 2|\vec{a}_k) x_{\alpha,\beta+1} \tag{3}$$

と表すことができる．我々の手法ではこのようにある時点での正解判定率と, 次の段階, さらに次の段階, という正解判定率の期待値を求めていき, 期待利得を最大化するように τ を求めていく．

例 1 p が $[0.5, 1]$ の一様分布に従うものとする．リクエスタが正しい判定によって得られるグロス効用 $r_C = 200$，誤った判定によって得られるグロス効用 $r_I = 0$ とする．さらに，各ワーカへの報酬 $c = 4$ とする．最大ワーカ数 $n = 11$ とする．

単にワーカ人数を 11 以下の定数（事前に定める）としてリクエスタは多数決で判定するものとすると, ワーカ数 1 のときは正解判定確率は 0.75 であり，リクエスタの収入の期待値は $0.75 \times 200 - 4 \times 1 = 146$ となる．同様に, $3, 5, 7, 9, 11$ 人のワーカの解答を聞いて判定するとリクエスタの期待収入はそれぞれ $150.5, 148.8, 144.7, 139.4, 133.4$ となる．すなわち, $\tau \equiv k$ という条件の下では $k = 3$ が最適となる．これは 11 人いるワーカを十分に活かしているとは言い難い．そこで, ワーカの解答を続けて聞くかどうかを途中の解答状況によって動的に決定し, 期待利得の向上を目指す．

途中段階の $k = \alpha + \beta$ 人目で $l = 1$ に α 票，$l = 2$ に β 票入っているとするとき,

$$\begin{aligned} P(\vec{a}_k) &= 2\int_{0.5}^{1} \left(\frac{1}{2} p^\alpha (1-p)^\beta + \frac{1}{2} p^\beta (1-p)^\alpha \right) dp \\ &= \int_{0}^{1} p^\alpha (1-p)^\beta dp = \frac{\alpha!\ \beta!}{(\alpha+\beta+1)!} \end{aligned}$$

であり,

$$\begin{aligned} & P(\vec{a}_k, a_{k+1} = 1) \\ &= 2\int_{0.5}^{1} \left(\frac{1}{2} p^{\alpha+1} (1-p)^\beta + \frac{1}{2} p^\beta (1-p)^{\alpha+1} \right) dp \\ &= \int_{0}^{1} p^{\alpha+1} (1-p)^\beta dp = \frac{(\alpha+1)!\ \beta!}{(\alpha+\beta+2)!} \end{aligned}$$

である．したがって,

$$P(a_{k+1} = 1|\vec{a}_k) = \frac{\alpha+1}{\alpha+\beta+2} = \frac{\alpha+1}{k+2}$$

である．よって，次のワーカの解答を聞くことにすると，次のワーカが $l = 1$ と解答する確率が $(\alpha+1)/(k+2)$ であり, $l = 2$ と解答する確率が $(\beta+1)/(k+2)$ であるから，$k+1$ 人目の時点で正解判定確率は

$$\frac{\alpha+1}{k+2} x_{\alpha+1,\beta} + \frac{\beta+1}{k+2} x_{\alpha,\beta+1} \quad (= x)$$

となる．もし $200x - 4(k+1) > 200x_{\alpha,\beta} - 4k$ であれば, $k+1$ 人目のワーカの解答を聞くべきと言える．

例えば, 最初の 3 人の解答が $(2, 1)$ 票に分かれたとき，ここで打ち切るかどうか判断するとする．ここで打ち切ったときの期待収入は $200x_{2,1} - 4 \times 3 = 125.5$ である．あと 2 人の意見を聞くとすると，ここから $(4, 1)$ 票になる確率は 0.4，$(3, 2)$ 票になる確率は 0.4, $(2, 3)$ 票になる確率は 0.2 であるから，この解答が正解になる確率は $0.4x_{4,1} + 0.4x_{3,2} + 0.2x_{2,3} = 0.752$ である．よって，リクエスタの期待収入は $0.752 \times 200 - 4 \times 5 = 130.5$ となり，3 人のワーカで打ち切るより期待収入が上がるため, ここでは打ち切らずに継続することとなる．

こうしてすべての停止時刻 τ について期待利得を調べると, 式 2 をみたす τ は「2 票差が生じたらそこでワーカの解答を打ち切る」であるとわかる．すなわち 2 票の差がつけば十分に正しい判定が可能とみてそこで判定を行い、1 票以内の差であれば引き続きワーカの意見を聞くのが最適となる．その結果, リクエスタの期待利得として 156.7 が達成される．

3. 自信を導入したワーカ数最適化問題

前章では，ワーカの能力の差は区別できないいう仮定のもとで，リクエスタは単純多数決投票によって最終的な解答を決定した．しかしながら，クラウドソーシングでの実際のワーカらには能力の差がある．文献 [Sakurai 13] では，解答と共に自信の有無を申告させるメカニズムが提案されている．ワーカに自信を申告させることで，リクエスタは効率的に正解判定を行うことが可能である．本論文では, こうしたワーカの自信を導入して票の重みづけにより正解判定率を改善させるモデルでも, ワーカ数を動的に決定することで期待利得の向上が可能であることを示す.

文献 [Sakurai 13] で提案されているメカニズムを簡単に説明する．ワーカ i は，自分の解答の自信度 (正解率) を $z_i (\geq 0.5)$ と認識しているものとする．一方，リクエスタはワーカの自信度が z であるときの平均正解率 $\nu(z)$ と，ワーカが認識する自信の分布の密度関数 $h(z)$ を知っているとする．

リクエスタは，タスクとなる問題と共に，プラン H とプラン L という 2 つの報酬プランをワーカに提示する．1 つの報酬プランは，正解と判定されたときの報酬と不正解と判定されたときの報酬の 2 つの報酬のペアで構成されている．ワーカは解答と共に報酬プランを申告する．この報酬プランは自信の値によって区分されている．

	正解	不正解
Plan H	a_H	b_H
Plan L	a_L	b_L

ワーカは自分の自信に応じて正直に報酬プランを決定することが最適であることが保証されている．ここでは，自信が高いワーカはプラン H, 自信が低いワーカはプラン L を選択するように設定されているとする．すなわち，ワーカ i は自分の自信 z_i から期待報酬を計算し，$z_i a_H + (1-z_i) b_H > z_i a_L + (1-z_i) b_L$ であればプラン H を選択し，そうでなければ，プラン L を選択することが最適戦略となる．

以下，自信を用いた重み付き多数決による判定によって正解判定率を向上させることができる．2 つの報酬プランを区分する自信の閾値を z_c とする．このとき，プラン H、プラン L を選択したワーカのそれぞれの正解率を p_H, p_L とすると

$$p_H = \int_{z_c}^{1} \nu(z)h(z)dz \Big/ \int_{z_c}^{1} h(z)dz$$

$$p_L = \int_{0.5}^{z_c} \nu(z)h(z)dz \Big/ \int_{0.5}^{z_c} h(z)dz$$

と算出される．

このとき，リクエスタはプラン H，プラン L を選択したワーカの解答にそれぞれ $\log \frac{p_H}{1-p_H}, \log \frac{p_L}{1-p_L}$ の重みをつけた多数決で正解を判定することとなる．これは，最尤推定法に基づく重みの決定方法である．

例 2 ワーカ数を $n = 11$ とし，リクエスタの正しい判定，誤った判定によるグロス効用をそれぞれ $r_C = 200, r_I = 0$ とする．ワーカの自信に関して，$\nu(z) = 0.8z + 0.1$ （ワーカはやや自信を過大評価する），$z \in [0.5, 1]$ に対して $h(z) \equiv 2$ (一様分布) とする．このとき，リクエスタは自信の閾値を $z_c = 0.75$ となるように報酬プランを次の表の通りに設定するとする．このとき，$p_H = 0.8$, $p_L = 0.6$ であることより，プラン H の票

	正解	不正解
Plan H	5	0
Plan L	4	3

は $\log(0.8/0.2) = 1.3863$ 票，プラン L の票は $\log(0.6/0.4) = 0.4055$ 票として重み付き多数決を行うこととなる．ワーカの自信は一様に分布し $z_c = 0.75$ であることより，ワーカの半数はプラン H，残り半数はプラン L を選択するため, リクエスタがワーカに払う平均報酬は $0.5 \times (0.8 \times 5 + 0.2 \times 0) + 0.5 \times (0.6 \times 4 + 0.4 \times 3) = 3.8$ である．

このとき, 事前に「必ず k 人目で打ち切る（すなわち $\tau \equiv k$)」と設定すると，$k = 1, 3, 5, 7, 9, 11$ での期待利得はそれぞれ $136.2, 149.6, 153.7, 153.8, 151.4, 147.5$ となり，ワーカ数は 7 が最適であるとわかる．しかし, ワーカ数を動的に「重み付き投票によって 2 票以上の差がついたら解答を打ち切る」とすると正解判定率は 0.919 と高い数値を維持しつつ, ワーカ数は平均 4.8 人と少なく抑えることが可能となり、期待利得として 165.6 という値を達成できる．

4. おわりに

クラウドソーシングは，Human computation の応用事例として，近年，脚光を浴びているアプリケーションの一つである．計算機と人間の能力を融合する上で，人間の能力をできる限り効率的に活用する必要がある．

その目的を達成する一つの提案として，本論文では，ワーカの能力に応じて，動的にワーカ数を最適化する手法を検討した．本論文では，最適停止問題を用いて，この問題を定式化することでリクエスタのワーカの能力に対する事前の予測と実際の能力が異なっていても，実際の能力を考慮してワーカ数の最適化を行うことを可能にした．

今後の研究の一つとして，より複雑なタスクを対象に，ワーカ数を最適化する手法の提案を行うことがある．

参考文献

[穴井 2000] 穴太 克則: タイミングの数理 最適停止問題, 朝倉書店 (2000)

[Ferguson 89] Ferguson, T. S.: Who solved the secretary problem?, *Statistical Science*, Vol. 4, pp. 282–289 (1989)

[John 10] Joseph, J. H. and Chilton, L. B.: The labor economics of paid crowdsourcing, *Proceedings of ACM Conference on Electronic Commerce (EC2010)*, pp. 209–218 (2010)

[鹿島 12] 鹿島 久嗣, 梶野 洸：クラウドソーシングと機械学習, 人工知能学会誌, Vol. 27, No. 4, pp. 381–389 (2012)

[Law 11] Law, E. and Ahn, L. V.: *Human Computation*, Morgan & Claypool Publishers (2011)

[Sakurai 13] Sakurai, Y., Okimoto, T., Oka, M., Shinoda, M., and Yokoo, M.: Quality-Control Mechanism utilizing Worker's Confidence for Crowdsourced Task, *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)* (2013)