2M4-OS-07a-5

クラウドソーシングを用いた順序統合

Rank Aggregation via Crowd Sourcing

石川真太郎[*1] 佐久間淳[*2]
Shintaro Ishikawa Jun Sakuma

[*1]筑波大学情報学群情報科学学類
College of Information Science, University of Tsukuba

[*2]筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻
Dept. of Computer Science, Graduate school of SIE, University of Tsukuba

The task of ranking a list of several alternatives based on one or more criteria is encountered in many situations, such as recommendation ranking of the books, restaurant ranking and etc. We consider a rank aggregation problem. To solve this problem, we usually use majority voting method or total score method often been used. However, these techniques don't consider the worker's performance into account. There exists some methods which consider all order of items, but they are much more complicated than majority voting. We propose efficient rank aggregation method that considers worker's performance using crowd sourcing. The experiment results show that our approachs achieves better accuracy and Spearman-distance than majority voting.

1. はじめに

複数人が幾つかの要素を並び替え，集団で 1 つの順序を生成する問題を考える．例えば，書店のお薦め本ランキングや，特定地域の飲食店ランキングなどがある．ランキングは，周囲の人間が今どんなことに注目しているか，選択肢の中でどの要素を選べば最も満足が高くなるか，などを知る手掛かりになることが多い．また近年ではビジネスにおいても，ビッグデータなど情報量が加速度的に増加しており，効率良く情報を集める方法として，推薦やランキングが注目されている．

ある要素の並べ替えにおいて最も単純なのは，評価尺度が単一で，その高順序のものから並べる手法である．これ以外に今までに用いられてきた手法としては，投票者がそれぞれ最も気に入った票に投票し，その票数順にランキングを生成する**多数決**がよく用いられている．また，それぞれの投票者がランキングを作成し，高順序に高い得点を与え，集団の和によってランキングを決定する合計得点方式（**Borda 得点法**など）も存在する．しかし，多数決では 2 位やそれより下位の順位に関する評価がされておらず，合計得点方式では要素数が増えるにつれて各投票者の計算コストが高くなってしまう．

このような計算コストの増加を防ぐ方法の 1 つとして，近年クラウドソーシングを使った，効率の良い集団予測を決定する手法が盛んに研究されている．クラウドソーシングは，画像のタグ付けや分類問題など，直感的に人間が答えやすい問題を解くのに適している．3 つ以上の要素を並び替える順序統合問題は直感的には答えにくいものの，問題を細かく分割することによって，複数の一対比較に答える問題に置き換えることができる．この 2 つの要素の比較である一対比較は，人間が直感的に答えやすい問題の 1 つであると言える．

またクラウドソーシングの問題では，解答者を能力によって選出するわけではないため，問題についての事前知識をある程度もつ優秀な解答者や，故意的に予測を正解から遠ざけようとする解答者が存在する場合がある．これらを解決するために，各解答者の能力をモデル化してクラウドソーシングの問題に適用し，解く手法が研究されている．

そして一対比較の集合から全順序へ統合する場合，一対比較の集合によって，無矛盾のものを生成するのが不可能となる場合がある．よって順序を統合する際に生じる矛盾を最小化しつつ，予測した一対比較をなるべく反映させた手法を考える必要がある．

更に順序統合問題では，他のタグ付け問題などとは異なり，与えられる問題が限定しにくく，問題から適切な特徴量を選択するのが非常に難しい．加えて，真の正解ランキングが明確で無い場合も多い．そこで特徴量を用いずに，教師なし学習で行えるような手法が求められる．

本稿では，上で述べたことを踏まえ，まずクラウドソーシングを用い，全順序を分割した一対比較を予測する．そこから得た結果を利用して，矛盾の少なくなるように効率的に順序統合を行う手法を提案する．

2. クラウドソーシングによる二値分類

ある問題に対し，クラウドソーシングを用いて，M 人の解答者（ワーカ）が存在する．各ワーカは，T 個の簡単な問題の中から任意に幾つかを選び解答する．ここでは，二値分類問題を扱う．

2.1 ワーカの能力モデル

当然であるが，各問題に対し，それぞれのワーカは事前知識にいくらか差が存在する．今回，ワーカ（問題の解答者）の能力を考慮するため"two-coin model" を用いてモデル化した．あるワーカ j について，感度 α^j，と特異度 β^j を式 (1) で定義する．

$$\begin{cases} \alpha^j = \Pr[y_i^j = 1 | y_i = 1] \\ \beta^j = \Pr[y_i^j = 0 | y_i = 0]. \end{cases} \tag{1}$$

教師付き学習において，真のラベルが 1 のとき，ワーカ j は真のラベルと同じラベル 1 を確率 α^j で返し，ラベル 0 を確率 $1 - \alpha^j$ で返す．真のラベルが 0 のとき，ワーカ j は真のラベ

連絡先: 石川真太郎，筑波大学情報学群情報科学学類，305-8573 茨城県つくば市天王台 1-1-1, shintaro@mdl.cs.tsukuba.ac.jp

ルと同じラベル 0 を確率 β^j で返し，ラベル 1 を確率 $1-\beta^j$ で返す．

2.2 ラベルとワーカ能力の推定

二値分類問題で扱う問題集合を $\mathcal{C}$ とし，T 個の問題（タスク）からなるとする．$\mathcal{C}$ には M 人のワーカが取り組むとし，各ワーカはそれぞれのタスクに 1 か 0 のラベル付けを行う．

各タスク i に対し，各ワーカ j がつけたラベルを y_i^j，さらに教師付き学習では真の正解ラベルを y_i，とする．また，集団（ワーカの集合）としての予測ラベル $\hat{y}_i$ が 1 となる確率を μ_i で表す．μ_i から予測ラベルを求めるための閾値として γ を用いる．

2.3 特徴量なしでワーカの能力を推定する手法

クラウドソーシングの問題を解く場合に，ワーカ毎の能力差が予測に大きな影響を与えることを述べた．各ワーカがそれぞれタスクにつけたラベルから，ワーカの能力を式 (1) で推定しつつ，集団の予測ラベルを求める手法が Raykar らによって提案されている (Alg.1)[2]．Raykar らの手法では，EM アルゴリズムを用いる．目的関数の対数尤度が式 2 で与えられ，この最大化問題を解く．

$$\mathbb{E}\{\ln \Pr[\mathcal{D}, \boldsymbol{y}|\boldsymbol{\alpha}, \boldsymbol{\beta}]\} = \sum_{i=1}^{N} \mu_i \ln pa_i + (1-\mu_i)\ln(1-p)b_i. \quad (2)$$

そして，最大化問題を解く際に，幾つかのパラメータ予測を同時に行う．各タスク i に対し，予測ラベルが真となる確率 μ_i は式 (3) で表される．

$$\mu_i = \frac{a_i p}{a_i p + b_i(1-p)}. \quad (3)$$

μ_i の各パラメータ a_i と b_i は，次式で表される．

$$a_i = \prod_{j=1}^{R} [\alpha^j]^{y_i^j}[1-\alpha^j]^{1-y_i^j}.$$

$$b_i = \prod_{j=1}^{R} [\beta^j]^{1-y_i^j}[1-\beta^j]^{y_i^j}.$$

ワーカの能力を表す感度 α^j および特異度 β^j は，それぞれ事前ベータ分布に従い，式 (4) と式 (5) で表される．

$$\alpha^j = \frac{a_1^j - 1 + \sum_{i=1}^{N} \mu_i y_i^j}{a_1^j + a_2^j - 2 + \sum_{i=1}^{N} \mu_i}. \quad (4)$$

$$\beta^j = \frac{b_1^j - 1 + \sum_{i=1}^{N} (1-\mu_i)(1-y_i^j)}{b_1^j + b_2^j - 2 + \sum_{i=1}^{N} (1-\mu_i)}. \quad (5)$$

なお，各ハイパーパラメータは,. $a_1^j = \alpha^j N$，$a_2^j = (1-\alpha^j)N$，$b_1^j = \beta^j N$，$b_2^j = (1-\beta^j)N$ で定義した．

$p = \Pr[y_i = 1]$ の推定は式 (6) で表され，これも事前ベータ分布に従う．

$$p = \frac{p_1 - 1 + \sum_{i=1}^{N} \mu_i}{p_1 + p_2 - 2 + N}. \quad (6)$$

p のハイパーパラメータは $p_1 = pN$，$p_2 = (1-p)N$ で表される．

式 (2) の最大化によって得られたパラメータ μ_i について，$\mu_i \geq \gamma$ であれば予測ラベルを $\hat{y}_i = 1$ とし，それ以外の場合は $\hat{y}_i = 0$ とする．

Algorithm 1 Expectation-Maximization Algorithm

Require: $\boldsymbol{y}^j (j \in M)$

Ensure: $\boldsymbol{\mu}$

```
Initialize μ_i by majority voting.
while E^(t+1) − E^(t) < threshold do
    M-step update α, β and p.
    E-step compute E{ln Pr[D, y|α, β]} and refine μ.
end while
```

3. クラウドソーシングによる順序統合手法の提案

3.1 クラウドソーシングによる順序統合

$\mathcal{X}_N = \{x_1, \ldots, x_N\}$ という要素集合に対して，真の順序を $O^* = x_{i_1} \succ \ldots \succ x_{i_N}$ で表す．また，順序 $x_p \succ x_q$ の順序ベクトル表現を $O_{p,q} = [x_p, x_q]$ で表す．順序統合問題 $\mathcal{R}$ は，真の順序 O^* と同じ順序 O を見つけることである．しかし，真の順序を複数の一対比較や要素の評価値から生成するとき，与えられる情報が足りない，与えられる情報に矛盾を含むなどの理由から O^* を見つけることが難しい場合がある．そのため実際には，O^* に近い順序 O を求める．本稿では，真の順序 O^* に対して順序 O の近さを，順序尺度である Spearman 距離 $d_{Spear}(O, O^*)$ により評価する．

クラウドソーシングは，人間が直感的に解くことのできるような小さな問題を大量に処理することができる．しかし，要素数が大きい順序統合のように大きな問題をそのまま処理するのには適していない．

本稿では，一対比較を用いて順序統合問題 $\mathcal{R}$ を解く方法を提案する．一対比較 $t_{p,q}$ とは,「要素 x_p と x_q のうち，どちらが前か？」というタスクである．ただし，x_p の方が前，すなわち $x_p \succ x_q$ であれば $t_{p,q} = 1$ を，x_q の方が前，すなわち $x_q \succ x_p$ であれば $t_{p,q} = 0$ を返す．

一対比較 $t_{p,q}$ は，ワーカ $j \in \{1, \ldots, M\}$ のつけたラベル $y_{p,q}^j$ から，前章で導入したクラウドソーシングを用いた二値分類により予測することができる．前章では，各タスクが単独の要素からなるため，予測ラベルを $\hat{y}_i$ としたが，ここでは，ペアワイズ要素に対して予測ラベルを決定するため，$\hat{y}_{p,q}$ と書く．$\hat{y}_{p,q} = 1$ なら $t_{p,q} = 1$，$\hat{y}_{p,q} = 0$ なら $t_{p,q} = 0$ とする．

本章では，各要素ペアの一対比較をクラウドソーシングを用いて予測し，比較結果 $t_{p,q}$ を用いて，最も矛盾の小さい順序 O を見つけるアルゴリズムを提案する．

3.2 既存手法

複数の部分順序が与えられているとき，$[x_p, x_q]$ かつ $[x_q, x_r]$ を，$[x_p, x_q, x_r]$ という様に矛盾のないように順に統合できれば，望む順序を得ることができる．この手法を**単純連結法**と呼ぶ．この手法では，総ての一対比較が無矛盾であれば，順序は一意に決まる．しかし，$[x_p, x_q]$ かつ $[x_q, x_r]$ かつ $[x_r, x_p]$ のように，循環，すなわち矛盾ある順序を生成してしまう場合には適用できない．

Gleich と Lim によって，行列分解を用いた手法 [1] が提案されている．この手法を**行列分解の手法**と呼ぶ．この手法では，2 つ要素 x_p と x_q の順序の差 $y_{p,q}$ を要素にもつ歪対称行列 $\boldsymbol{Y}$ で表現する．$\boldsymbol{Y}$ は疎行列でも構わない．この $\boldsymbol{Y}$ について，ニュークリアノルム $||\boldsymbol{Y}||_*$ を制約として低ランク近似により行列を補完し，その値より順序を得る．この手法では，x_p と x_q に評価値のような連続値を用いるが，一対比較のように，結果が二値で与えられるようなことを想定していない．

Algorithm 2 Rank aggregation via Queue Algorithm

Require: $\hat{\boldsymbol{y}}$
Ensure: O
for all $p, q \in \{(1, \ldots, N), p \neq q\}$ **do**
  if $y_{p,q} = 1$ **then**
    $\Omega \leftarrow [x_p, x_q]$
  else
    $\Omega \leftarrow [x_q, x_p]$
  end if
end for
for all $[x_p, x_q] \in \Omega$ **do**
  $\mathcal{Q} \leftarrow enqueue([x_p, x_q])$.
end for
while ($\mathcal{Q}$ の先頭にある順序 O の長さ)$= N$ **do**
  $O \leftarrow dequeue(\mathcal{Q})$
  $x_{tail} \leftarrow$ (O の最下位要素)
  for all r s.t. $[x_{tail}, x_r] \in \Omega, I_O \cap r = \phi$ **do**
    $O' \leftarrow O.append(x_r)$
    $\mathcal{Q} \leftarrow enqueue(O')$
  end for
end while
$O_{fin} \leftarrow \arg\min_{O'' \in \mathcal{Q}} d_{Kendall}(O'', \Omega)$
O_{fin} が複数ある場合はランダムに順序 O'_{fin} を選ぶ.

そこで，一対比較が与えられ，それらに矛盾を含むような場合にも，なるべく一対比較と統合する順序の矛盾が小さくなるような順序統合を行う手法を提案する．

3.3 キューを用いた順序統合

提案手法では，統合する順序と矛盾する一対比較数を増やさないように，キューを用いて，繋がりうる可能性のある要素を順序 O に追加し，候補を列挙する．そして，最終的に残った候補の中から，与えられた一対比較との矛盾が小さいものを統合順序 O_{fin} として扱う．

まず，観測された $\hat{y}_{p,q}(p, q \in \{1, \ldots, N\}, p \neq q)$ を一対比較 $t_{p,q}$ と見なし，その順序を順序ベクトルの形で集合 Ω に追加する．その後，集合 Ω の全要素を，キュー $\mathcal{Q}$ にエンキューする．

次に，$\mathcal{Q}$ の先頭にある順序 O の長さが要素数 N に達し，$\mathcal{X}_N$ を総て含む順序 O'' が生成されるまで，次の操作を繰り返す．ただし，順序 O の要素のインデックス集合を I_O，x_{tail} は O の末尾を，順序 O から要素 x_{tail} を除いた順序を O_{-1} で表す．

要素を順序に繋げていくとき，矛盾する一対比較数を減らすために Ω を利用する．$[x_{tail}, x_r] \in \Omega$，かつ $I_O \cap r = \phi$ を満たす要素 x_r があれば，順序 O に加え，新たに順序 O' を生成，$\mathcal{Q}$ にエンキューする．$\mathcal{Q}$ の先頭にある順序 O の長さが N に達していれば，$\mathcal{Q}$ に含まれる総ての順序の長さも N に達していることは，キューの性質からも明らかである．

長さが N に達した候補となる順序 $O'' \in \mathcal{Q}$ の中から，Kendall 距離が最小となる順序を最終的な順序 O_{fin} として表す．Kendall 距離は，同じ要素対集合からなる 2 つの順序対について，対称の対 $x_p, x_q \in \{1, \ldots, N\}$ の中で順序が不一致なものの対の数を表す．$d_{Ken}(O'', \Omega)$ は，順序 O'' から生成した順序対の集合と Ω との Kendall 距離を表す．

4. 実験

本章では，まず複数のワーカがラベル付けした一対比較のタスクから，クラウドソーシングを用いて集団の一対比較を推定し，同時にワーカの能力も推定することで，真のラベルや特徴量がない場合でも，多数決を使った集団の一対比較予測よりも精度が高くなることを示す．

そして，提案手法により予測した 2 つの一対比較結果から，真の順序に近い順序が統合でき，このときもワーカの能力を推定した場合の方が精度が高くなることを示す．また行列分解の手法が，歪対称行列 $\boldsymbol{Y}$ の要素に一対比較のような二値が与えられる場合を想定していないため，相対的に精度が低くなることを実験で示す．

設定　集団としての順序を求めるとき，真の正解がある場合とない場合が存在する．実際には真の正解がない場合の方が多いが，今回は他の手法と提案手法を比較するために，真の正解がある場合を用いて，ランキングの予測を行う．

要素数が N 個あるとき，各ワーカに N 個の要素の並び替え問題を出題するよりも，$N(N-1)/2$ 個の一対比較の方が直感的で答えやすくなる．そこで，データセットも一対比較を問う形で収集した．

本実験では，まず一対比較の予測について，クラウドソーシングを用いてワーカの能力を考慮した場合（**クラウドソーシング**とする）の結果と，考慮しないで多数決を用いた場合（**多数決**と呼ぶ）の結果とを精度によって比較する．そしてキューを用いた順序統合アルゴリズムによって，それぞれ予測した一対比較結果とから順序をキューを用いた手法（**提案手法**と呼ぶ）により統合する．統合した順序と真の正解ランキング（**真順序**と呼ぶ）との誤差を Spearman 距離によって比較する．

また行列分解を用いた手法（**行列分解**と呼ぶ）でも真順序との誤差を Spearman 距離によって計算，比較した．この手法において，二値で与えられる場合を想定していないことを示すために，歪対称行列 $\boldsymbol{Y}$ の要素には，二つの要素 x_p と x_q の一対比較の結果 $t_{p,q} \in \{0, 1\}$ を与えた．

データセット　多数決による手法と比較するために，Lancers タスクを利用して 2 種類の一対比較データセットを新しく生成した．

1 つ目は，2012 年に Yahoo! JAPAN で検索された人物のうち上位 1〜15 位に並べる問題であり，これを**問題 1** とする．一対比較が全部で 105 問あり，ワーカ 300 人がその総てに解答した．2 つ目は，お菓子を発売された年の早い順に並べる問題であり，これを**問題 2** とする．一対比較が全部で 105 問，213 人のワーカが解答した．

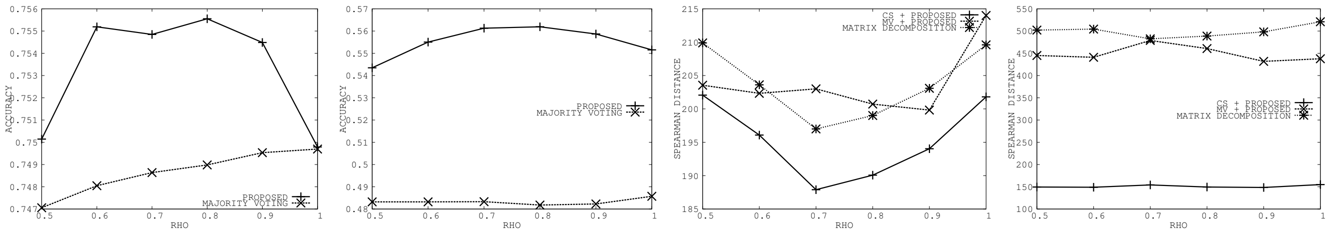
問題 1 と問題 2 に答えたワーカから，それぞれランダムに 200 人選んで実験を行った．

一般に，クラウドソーシングで与える問題の難しさを測ることは難しい．そこで，各一対比較の難しさを表す 3 つの尺度を，表 1 に示す．それぞれ，与えた一対比較毎の平均正答率，最も簡単な一対比較の正答率，最も難しい一対比較の正答率を示す．

表 1: 与えられた問題の一対比較の難しさ

	Average	Easiest	Hardest
問題 1	62.97%	97.50%	9.50%
問題 2	49.70%	97.00%	2.00%

実際には，一対比較で回答した場合においても，ワーカ 1 人



(a) 問題 1 における一対比較正答率

(b) 問題 2 における一対比較正答率

(c) 問題 1 における真順序との Spearman 距離

(d) 問題 2 における真順序との Spearman 距離

図 1: 一対比較の正答率（横軸: ワーカの回答率, 縦軸: 正答率）と統合した順序と真順序の正答率

あたりの負担は大きくなってしまう．そこで，それぞれのワーカが答える問題数の割合 ρ を 0.5 から 1.0 まで 0.1 ずつ変化させ，統合した順序の精度とのトレードオフを評価した．

まず，クラウドソーシングと多数決における精度を求め，その後，順序統合における誤差を計算した．実験はそれぞれ 1000 回行い，結果としてその平均を用いた．

評価指標　クラウドソーシングによる T 個の一対比較予測を評価するために，式 (7) で正答率の平均精度を求めた．

$$\mathrm{ACC} = (1/T)\sum_{i=1}^{T}\delta_{y_i,\hat{y}_i}. \tag{7}$$

なお，$\delta_{i,j}$ は $i=j$ のとき 1，$i \neq j$ のとき 0 をとる．ACC は大きな値をとる程，予測精度が高いことを示す．

また統合した順序に対する評価を行うために，順序 O_x と O_y の Spearman 距離を式 (8) で定義する．O_x での対象 i の順位を $r_{(x,i)}$ で表す．

$$d_{Spear}(O_x, O_y) = \sum_{i=1}^{N}(r_{(x,i)} - r_{(y,i)})^2. \tag{8}$$

Spearman 距離は，二つのランキングが完全に一致するとき最小値 0，逆順序のとき最大値をとる順序尺度である．そこで，真の正解ランキングとの Spearman 距離を用いて，予測したランキングの誤差が小さくなることを示す．

結果と考察　問題 1，問題 2 のそれぞれについて，ワーカの割合と一対比較の予測結果における精度を図**??**に，ワーカの割合と順序統合したときの Spearman 距離における誤差を図 1 に示す．

ρ を 0.5〜1.0 まで 0.1 ずつ変化させ，ワーカの解答するラベル数を変化させた．一対比較の精度について，問題 1 ではクラウドソーシングと多数決との差が 1%未満となった．これに対し問題 2 では，5〜6%程良い結果が得られた．このことから，ワーカの能力を予測することは簡単な問題を解く場合には勿論，難しい問題の場合には特に，誤差の上昇を抑えることができると考えられる．

提案手法により統合した順序 O_{fin} と真順序 O^* との Spearman 距離についても，クラウドソーシングが多数決よりも良い精度を示した．このことから，順序統合問題でも，ワーカの能力を予測することで多数決よりも真順序に近い順序を生成できることが確認できた．

行列分解は，どちらの問題に対しても多数決と同程度の誤差を示した．このことから，行列分解の手法では，一対比較のような二値を想定しておらず，このような値を用いる場合には，二値を連続値にするような変換が必要であると考えられる．

ワーカの解答割合 ρ について，提案手法でも，行列分解の手法でも，割合毎による精度の変化はあまり見られなかった．

提案手法は，問題の難易度や扱う要素数が増えると計算量が非常に大きくなってしまうことがわかった．様々な問題を扱うために，最適化のような，より効率的なアプローチが必要になる．

5.　まとめ

本稿では，クラウドソーシングを用いて，ワーカ間の能力差を考慮し，予測した一対比較を用いて，真の順序との Spearman 距離が近くなるような順序を統合ための手法を提案し，多数決よりも精度，誤差が改善されることを示した．また，異なる 2 つのデータセットに対し，特徴量を用いなくても，多数決よりも性能が良いことを示すことができた．

今後の課題として，ワーカの能力を予測し，最適化を用いて順序統合を行う手法，各ワーカが一対比較に対してラベルをつけ，そのラベルを用いた順序の推薦などが考えられる．

謝辞

本研究は，最先端研究開発プログラム「超巨大データベース時代に向けた最高速データベースエンジンの開発と当該エンジンを核とする戦略的社会サービスの実証・評価」の助成を受けました．

参考文献

[1] David F. Gleich and Lek-Heng Lim. Rank aggregation via nuclear norm minimization. *CoRR*, 2011.

[2] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Anna Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. Supervised learning from multiple experts: Whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.