

MOOCにおける相互評価のための適応的な割り当て手法

大橋 英明[†] 清水 敏之^{††} 吉川 正俊^{††}

[†] 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

^{††} 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: [†]ohashi@db.soc.i.kyoto-u.ac.jp, ^{††}{tshimizu,yoshikawa}@i.kyoto-u.ac.jp

あらまし 近年、大規模公開オンライン講座 (Massive Open Online Courses) の中で受講者間の相互評価を利用した採点が実施されている。相互評価は受講者が数万人に及ぶ MOOC で採点を行うための手法として有望視されているが、一方で相互評価を利用すると課題提出率が低下するという報告も存在する。相互評価に不信感を持つ受講者が多く存在することが原因の一つと言われており、相互評価の品質を向上させる手法が課題提出率の向上に有効だと考えられる。本研究は答案の割り当てによる品質の向上に焦点を当て、相互評価の過程でドロップアウトする受講者を考慮した上で、各受講者の採点数と被採点数の差の絶対値の合計をなるべく小さくすることと、各受講者の答案に割り当てられる採点者の採点能力値の平均をなるべく等しくすることを目的とした適応的な答案割り当て手法を提案する。

キーワード MOOC, 相互評価, タスク割り当て

1. はじめに

近年、インターネット上で受講できる大規模公開オンライン講座 (Massive Open Online Courses) が注目を集めている。MOOC では著名な講師の講義にもかかわらず、無料で公開されているものが多数存在し、時には一つの講義の受講者が数万人に及ぶこともある [16]。このとき、講師や TA (teaching assistant) だけで全ての採点を行うことは困難だと考えられる。

受講者間で互いの答案を採点する相互評価 (peer assessment) は受講者が提出した大量の答案を評価するための手法として有望視されている [11], [17]。しかし、相互評価が実施される課題の提出率は他の課題の提出率と比べて低くなることが報告されており、これは相互評価での採点に不信感を持つ受講者が課題に取り組みないことが原因の一つだと考えられる [15]。採点に不信感を抱かせないためには、相互評価の採点品質を向上させる手法が重要だと考えられる。そのような手法として、採点内容に対して自動採点を行い (automated meta-reviewing)、採点者へ採点内容の修正・改良を促す手法 [18]、被採点者に採点者の採点内容についてフィードバックさせる手法 [7] などが提案されている。また、一つの答案につき複数の採点者が存在する場合、各受講者の採点能力を推定し、個々の採点結果を統合するような手法も提案されている [4], [17]、これらの手法は個々の答案の採点品質の向上のみに焦点を当てているが、採点者と答案の割り当ても相互評価の品質の向上に不可欠だと考えられるため、本研究は採点者と答案の割り当て手法に焦点を当てる。

本研究は相互評価の過程でドロップアウトする受講者の存在を考慮した適応的な割り当て手法を考える。特に、以下の二点を目的とした採点者と答案の割り当て手法を提案する。

- (A) 各受講者の採点数と被採点数の差の絶対値の合計をなるべく小さくする
- (B) 各受講者の答案に割り当てられる採点者の採点能力値の平均

均をなるべく等しくする

目的 (A) の意義について説明する。仮に相互評価の過程でドロップアウトする受講者が存在せず、すべての受講者があらかじめ指定された通りに採点を行うのであれば、全受講者の採点数と被採点数の差を 0 にするよう事前に割り当てることができる。しかし実際には、あらかじめ指定された通りに答案を採点せず、ドロップアウトする受講者が存在する [2], [9]。受講者がドロップアウトする場合、ドロップアウトする受講者にあらかじめ割り当てられた答案のうちいくつかは採点されないという望ましくない事態が起こり得る。このとき、他受講者の被採点数が採点数より少なくなる場合があり、これは真面目に採点を行なった受講者が十分なフィードバックを受けられない可能性を示している。以上のような事態を避けるため本研究は (A) を目的とした適応的な割り当て手法を考える。

目的 (B) の意義について説明する。各受講者の答案に割り当てられる採点者の採点能力値の平均にばらつきがあるとき、特定の受講者の答案に採点能力の低い受講者ばかりが割り当てられ、その受講者が採点の結果に不信感を抱く、といったことが起こり得る。そこで、採点品質に不平等が発生しないように本研究は (B) を目的とした割り当てについても考慮する。

本研究では、まず (A) を目的とした割り当てアルゴリズムを提案する。続けて、提案したアルゴリズムを拡張して (A) と (B) を同時に目的とした割り当てアルゴリズムを提案する。実験では、Canvas Network が公開している相互評価のデータ [14] と人工データを用いて、提案手法の有用性を示す。

本論文の構成を以下に示す。2 節では関連研究の紹介を行う。3 節では本研究の問題設定を述べ、(A) を目的としたアルゴリズムと、(A) と (B) を目的としたアルゴリズムを示す。4 節では実データと人工データを用いて、提案したアルゴリズムの有用性を示す。5 節ではまとめと今後について述べる。

2. 関連研究

2.1 個々の答案の採点品質の向上に関連した研究

近年、クラウドソーシングに関する研究が精力的に行われているが、クラウドソーシングの研究と相互評価の研究は非常に関連が深い [21]. 特に、クラウドソーシングの文脈で行われている品質管理の研究を相互評価の採点結果の統合に応用した研究は数多く存在する [5], [8], [17], [19], [20].

また、相互評価の採点品質の向上を目的とした他の手法としては、採点内容に対して自動採点を行い (automated meta-reviewing), 採点者へ採点内容の修正・改良を促す手法 [18], 被採点者に採点者の採点内容についてフィードバックさせる手法 [7] などが提案されている. 相互評価の際に用いられるルーブリック (採点基準) の品質を高めることは、そのまま採点品質の向上に直結するため、ルーブリックの効果を検証する研究も存在する [3], [11].

2.2 答案と受講者の割り当てに関連した研究

クラウドソーシングにおける割り当て手法は既に数多く提案されている [1], [6], [13], [22]. 一方で、相互評価における割り当て手法はあまり提案されていない.

Est'vez-Ayres らはドロップアウト等による採点の不足を避けるための割り当てメカニズムを提案し、シミュレーションでその有用性を確認した [10]. 本研究の目的 (A) と類似した目的を掲げているが、本研究と異なり、自身の採点数が被採点数よりも多くなることを許容する受講者を仮定しており、本研究の提案手法と比較を行うことは困難である. また実データを用いた有用性の評価も行っていない.

Han らは Longest Processing Time と呼ばれるアルゴリズムを応用して答案ごとに割り当てられる採点者の能力に大きな差が生まれぬような割り当て手法を提案している [12]. この研究では、本研究のように採点能力が所与であると仮定し、(B) を目的とした割り当てを行っているが、受講者のドロップアウトを考慮していない.

我々の知る限り、ドロップアウトする受講者の存在を考慮して (A) と (B) の両者を目的とした研究は本研究が最初である.

3. 問題設定・アルゴリズム

本研究は「(A) 各受講者の採点数と被採点数の差の絶対値の合計をなるべく小さくする」「(B) 各受講者の答案に割り当てられる採点者の採点能力値の平均をなるべく等しくする」という二つの目的に対して、(A) のみを目的とした割り当てアルゴリズムと、(A) と (B) を同時に目的とした割り当てアルゴリズムを提案する. 受講者の識別子を $i \in \{0, \dots, n-1\}$, 受講者の採点状況を管理する変数 g_{ij} とし、受講者 i が受講者 j の答案の採点を完了したとき $g_{ij} = 1$, 採点中のとき $g_{ij} = 0.5$, 採点していないとき $g_{ij} = 0$ とする. ただし、開始した採点を途中で放棄する受講者は存在しない、すなわち g_{ij} は最終的に 0 か 1 のどちらかの値を取るものとする. また、受講者 i の採点能力値を $a_i \in [0, 1]$ とする. ここでは簡単のために採点能力値 a_i が全ての答案に対して共通すると考えているが、受講

者や答案の性質によって採点の有用さが変化するような状況 (例えば、極端に優秀な受講者は初学者にとって有用な採点できない、など) も考えられる. 本研究が提案するアルゴリズムはそのような状況にも対応可能である. 採点能力値 a_i を推定する方法として、受講者の学習状況や過去の採点状況などを利用した推定手法が考えられるが、本研究では既存研究 [12] と同様に a_i の推定を行わず、 a_i は事前に与えられると仮定する. t 番目 (以降、時刻 t と呼ぶ) に採点を開始する受講者の識別子を $i^t \in \{0, \dots, n-1\}$ とする. すなわち、全受講者の採点回数を l 回とすると、受講者の採点開始順序は $\langle i^0, \dots, i^{l-1} \rangle$ で与えられる. 時刻 t に受講者 i^t に割り当てられる答案を作成した受講者の識別子を $p^t \in \{0, \dots, n-1\}$ とする. また、採点時間の最大値を整数 r_{max} で与え、各採点の採点時間を $\langle r^0, \dots, r^{l-1} \rangle (r^t \in \{0, \dots, r_{max}\})$ とする. ただし、時刻 t に開始された採点は時刻 $t+r^t$ まで続き、その間受講者 i^t は他の答案を採点することができないものとする.

3.1 (A) を目的とした割り当て

(A) を目的とした割り当ては、すべての採点が終了した時点における以下の目的関数の最小化を目標とする.

$$\sum_i \left| \sum_j g_{ij} - \sum_j g_{ji} \right| \quad (1)$$

$\sum_j g_{ij}$ は受講者 i の採点数、 $\sum_j g_{ji}$ は受講者 i の被採点数を表しており、目的関数 (1) の最小化は目的 (A) を目指していることがわかる. 本研究は各時刻で採点数と被採点数の差が最も大きな受講者の答案の採点を優先するアルゴリズムを提案する. ある時刻 t での割り当てアルゴリズムの詳細は以下の通りである.

Algorithm 1 AdaptiveAllocationforPeerGrading

INPUT: reviewer i^t

INPUT: current grading state $g_{ij} \in \{0, 0.5, 1\}$

OUTPUT: allocation p^t

```
1:  $U \leftarrow \{\}$  ▷ 受講者  $i^t$  が未採点の受講者集合を格納する
2: for  $j \leftarrow 0$  to  $n-1$  do
3:   if  $g_{i^t j} == 0$  then
4:      $U.add(j)$ 
5:   end if
6: end for
7:  $p^t \leftarrow \arg \max_{k \in U} \sum_l g_{kl} - \sum_l g_{lk}$ 
8: return  $p^t$ 
```

入力は、時刻 t に採点を行う受講者の識別子 i^t と全受講者の採点状況 g_{ij} で、出力は受講者 i^t に割り当てられる答案を作成した受講者の識別子 p^t である. ただし、時刻 t に採点を行った受講者は時刻 $t+1$ から時刻 $t+r^t$ まで採点に取り組むことができず、その間アルゴリズム 1 の入力となり得ない. アルゴリズム 1 は受講者 i^t が未採点の受講者の中から採点数と被採点数の差が最も大きい受講者の答案を受講者 i^t に割り当てるアルゴリズムである. 図 1 を用いて、アルゴリズム 1 の直感的な説明を行う. ただし、この図では 5 人の受講者が存在し、 $r_{max} = 0$,

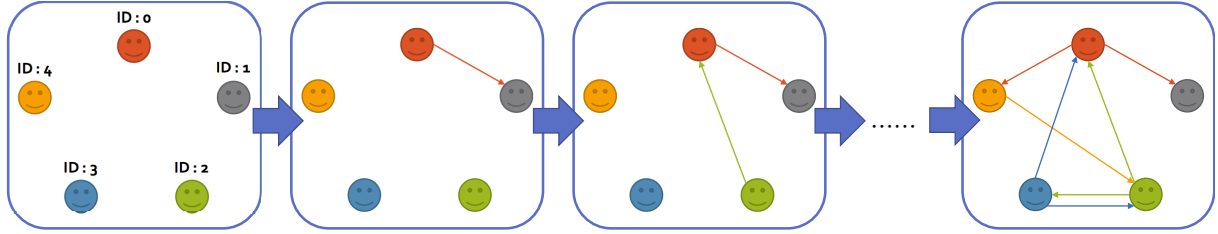


図 1: (A) を目的とした割り当てアルゴリズムの挙動例。採点した受講者から採点された受講者に実線の矢印が引かれている。ただし、採点時間は考慮せず、採点開始順序は $\langle 0, 2, 4, 0, 3, 2, 3 \rangle$ とする。

つまり採点時間を考慮せず、採点開始順序は $\langle 0, 2, 4, 0, 3, 2, 3 \rangle$ とする。初期状態では全受講者の採点数と被採点数の差は 0 なので、受講者 0 にはランダムに答案が割り当てられる。この図では受講者 1 の答案が選択されている。受講者 0 が受講者 1 の答案を採点した後、直前に採点を行った受講者 0 の答案が受講者 2 に割り当てられる。続けて、受講者 0 の採点数と被採点数の差は 0 に戻るが、受講者 2 の採点数と被採点数の差は増加するため、受講者 2 の答案が優先的に割り当てられる。図 1 では、以上のような割り当てが繰り返されており、アルゴリズム 1 が目的関数 (1) の最小化を目指したアルゴリズムとなっていることが直感的に理解できる。

3.2 (A) と (B) を目的とした割り当て

(A) と (B) を同時に目的とした割り当ては、すべての採点が終了した時点における以下の目的関数の最小化を目標とする。

$$\sum_i \left| \sum_j g_{ij} - \sum_j g_{ji} \right| + \lambda \cdot \sum_{i \in I'} \left| \sum_j \left(\sum_{i \in I'} a_j \cdot g_{ji} / \sum_j g_{ji} \right) / |I'| - \sum_j a_j \cdot g_{ji} / \sum_j g_{ji} \right| \quad (2)$$

ただし、 I' は被採点数が 1 以上の i の集合を指し、 $|I'|$ はその集合の大きさを表すものとする。目的関数 (2) の前半部分 $\sum_i |\sum_j g_{ij} - \sum_j g_{ji}|$ は目的関数 (1) と同様である。目的関数 (2) の後半部分 $\sum_{i \in I'} \left| \sum_{i \in I'} \left(\sum_j a_j \cdot g_{ji} / \sum_j g_{ji} \right) / |I'| - \sum_j a_j \cdot g_{ji} / \sum_j g_{ji} \right|$ が目的 (B) を考慮した部分となっている。これは各答案に割り当てられた採点者の採点能力値の平均 $\sum_j a_j \cdot g_{ji} / \sum_j g_{ji}$ を求め、受講者 $i \in I'$ について平均を取った値 $\sum_{i \in I'} \left(\sum_j a_j \cdot g_{ji} / \sum_j g_{ji} \right) / |I'|$ と、各答案に割り当てられた受講者の採点能力値の平均 $\sum_j a_j \cdot g_{ji} / \sum_j g_{ji}$ の差の絶対値を取り、受講者 $i \in I'$ について総和を取ったものとなっている。この値が小さくなるにつれ、各受講者の答案に割り当てられる採点者の採点能力値の平均は互いに近い値を取る。なお、 λ は非負のパラメータであり、目的 (A) を目的 (B) よりも重視したい場合は λ を小さくし、逆の場合は λ を大きくすれば良い。ある時刻 t での (A) と (B) を目的とした割り当てアルゴリズムとして、以下のアルゴリズムを提案する。

Algorithm 2 ModifiedAdaptiveAllocationforPeerGrading

INPUT: reviewer i^t

INPUT: current grading state $g_{ij} \in \{0, 0.5, 1\}$

OUTPUT: allocation p^t

```

1:  $U \leftarrow \{ \}$  ▷ 受講者  $i^t$  が未採点の受講者集合を格納する
2: for  $j \leftarrow 0$  to  $n - 1$  do
3:   if  $g_{i^t j} == 0$  then
4:      $U.add(j)$ 
5:   end if
6: end for
7:  $p^t \leftarrow \arg \max_{k \in U} \sum_l g_{kl} - \sum_l g_{lk} - \lambda \cdot \left| \sum_l a_l / n - \left( \sum_l a_l \cdot g_{lk} + a_i \right) / \left( \sum_l g_{lk} + 1 \right) \right|$ 
8: return  $p^t$ 

```

入力は、時刻 t に採点を行う受講者の識別子 i^t と全受講者の採点状況 g_{ij} と全受講者の採点能力値 a_i で、出力は受講者 i^t に割り当てられる答案を作成した受講者の識別子 p^t である。アルゴリズム 2 では、アルゴリズム 1 で考慮した採点数と被採点数の差から、全受講者の採点能力値の平均 $\sum_l a_l / n$ と、既に割り当てられている採点者の採点能力値と採点者 i の採点能力値の平均 $(\sum_l a_l \cdot g_{lk} + a_i) / (\sum_l g_{lk} + 1)$ の差の絶対値を引いた値が最も大きな受講者の答案の割り当てが優先されるようなアルゴリズムとなっている。ただし、各答案に割り当てられた採点者の採点能力値の平均を受講者 $i \in I'$ について平均した値 $\sum_{i \in I'} \left(\sum_j a_j \cdot g_{ji} / \sum_j g_{ji} \right) / |I'|$ は、採点が終わった後でなければ求めることができないので、アルゴリズム 2 では、この値を近似する値として全受講者の採点能力値の平均 $\sum_l a_l / n$ を用いている。また、本研究では採点能力値 a_i を全受講者の答案に対して共通な値として考えているが、受講者・答案によって採点能力値が異なる場合にも容易に拡張することが可能である。具体的には、受講者 i が受講者 j の答案を採点する際の採点能力値 a_{ij} を考え、アルゴリズム 2 の中で受講者 i が受講者 j の答案を採点する際の採点能力値として、 a_i の代わりに a_{ij} を用いれば良い。

4. 実験

実データと人工データの二種類のデータを用いて、提案手法の有用性を示す。ただし、利用する実データには以下の二点の問題がある。

- 各受講者の採点回数を読み取ることはできるが、各受講者

がどの受講者の答案を採点したのかを読み取ることはできない

- 厳密な採点開始順序・採点時間を読み取ることができない

よって、本実験では実データに対していくつかの補完を行う。まず実データと人工データの説明を行い、続けて比較手法、実験結果について述べる。

4.1 実データ

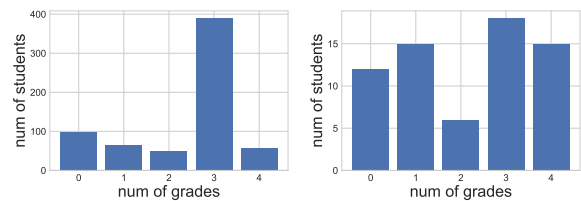
実データには Canvas Network が公開しているデータを利用する。本実験では特に、授業の識別子が 770000832960949, 課題の識別子が 770000832930436 のデータを基にしたデータ (実データ 1 と呼ぶ) と、授業の識別子が 770000832945340, 課題の識別子が 770000832960431 のデータを基にしたデータ (実データ 2 と呼ぶ) を用いる。具体的には, *submission_comment_fact* というテーブル中の提出物の識別子, 提出物にコメントした者の識別子, 各コメント量を利用する。提出物にコメントした者は厳密には採点者と異なる可能性があるが, 本研究では提出物にコメントした者の識別子を採点者の識別子として扱う。本実験ではコメント量を用いて採点能力値を設定することとし, 各採点におけるコメント量を採点者の識別子ごとに集約して平均を取り, この値が小さい採点者の採点能力値から順に 0.2, 0.4, 0.6, 0.8, 1.0 とした。ただし, 各採点能力値を持つ採点者の数となるべく等しくなるように調整した。Canvas Network のデータには, 提出物を出した受講者 (被採点者) の識別子と提出物を採点した受講者 (採点者) の識別子を紐付ける情報が存在しないため, どの受講者がどの受講者の答案を採点したのか読み取ることができない。これはデータの公開に伴う匿名化が原因だと考えられる。よって, 実際に行われた採点における目的関数 (1) の値と目的関数 (2) の値を算出することは不可能であるため, 本研究では, 提案手法を用いたシミュレーションの結果と既存手法 (後述) を用いたシミュレーションの結果を比較し, 提案手法の有用性を主張する。また, Canvas Network のデータからは各採点の厳密な開始時刻・終了時刻を読み取ることができない。そこで本研究は以下の二通りの方法で, 採点開始順序 $\langle i^0, \dots, i^{l-1} \rangle$ と採点時間 $\langle r^0, \dots, r^{l-1} \rangle$ を補完した。

4.1.1 採点コメントの作成日時に基づく補完

一つ目の方法は, *submission_comment_dim* というテーブル中にある, 採点コメントが作成された日時を利用した方法である。この値が小さいものから順番に採点者の識別子を並び替え, 採点開始順序とする。ただし, 採点時間に関するデータが得られないため, $r_{max} = 0$, すなわち採点時間を考慮しないものとする。この方法に基づき補完されたデータは本実験で用いるデータの中で最も現実のデータに依存したものとなる。

4.1.2 採点遷移モデルに基づく補完

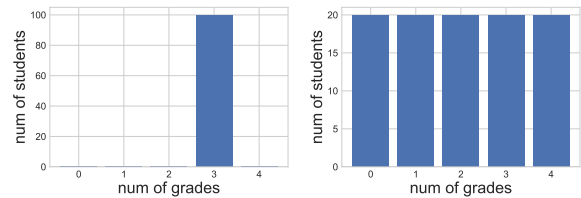
まず, 採点者ごとの採点数を求める。図 4.4 は実データ 1 と実データ 2 について採点数ごとに採点者数をプロットしたものである。ただし実データ 1 では, 一人の採点者が採点終了間際に 25 回の採点を行っており, これは講師または TA が被採点数が不足した受講者の答案を採点した形跡だと考えられる。採点遷移モデルに基づきデータを補完する際は, その採点者の採点数



(a) 実データ 1

(b) 実データ 2

図 2: 実データの採点数ごとの採点者数



(a) 偏在データ

(b) 均一データ

図 3: 人工データの採点数ごとの採点者数

を最頻値の 3 に置き換えた。また, *submission_comment_fact* からは採点を行わない採点者に関する情報を陽に得られないが, 提出物の識別子の種類数から, 採点者の識別子の種類数を引いた数値が, 課題を提出しているにも関わらず採点を行っていない受講者の数にあたると考え, その値を採点数 0 の採点者数として用いた。

採点遷移モデルでは, 採点者ごとの採点数を用いて, 確率 P で採点者 i^{t+1} を直前の採点者 i^t となるべく同じになるように, 確率 $1 - P$ で採点者 i^{t+1} をランダムに選択し, 採点開始順序を生成する。例えば, $P = 0$ のとき, 採点者 i^{t+1} は直前の採点者 i^t と関係なくランダムに選択される。 $P = 1$ のとき, 採点者 i^{t+1} は直前の採点者 i^t となるべく同じになるように選択される。また採点時間については, 採点時間がすべて 0 となる場合と, 採点時間が最小値が 0, 最大値が 10 となるよう一様ランダムに生成される場合の二通りのデータを生成する。

4.2 人工データ

以下の二種類の人工データを用いた。一つ目は採点数が 3 の採点者のみが存在するようなデータ (偏在データと呼ぶ) で, 二つ目は採点数が 0 から 4 の採点者数が均一に存在するようなデータ (均一データと呼ぶ) であり, どちらも全受講者数が 100 人となるようにした (図 3 参照)。前者では, ドロップアウトする受講者が存在せず, 採点数が一つに集中する場合を想定しており, すべての受講者があらかじめ講師に指定された数だけ採点を行うような状況と捉えられる。後者では, 採点数がばらばらになった状況を想定している。また, 図 4.4 と図 3 から, 実データ 1 と偏在データのヒストグラム, 実データ 2 と均一データのヒストグラムが各々類似していることが読み取れる。偏在データは実データ 1 の傾向を持つデータの極端な例, 均一データは実データ 2 の傾向を持つデータの極端な例と捉えることができる。採点開始順序と採点時間は 4.1.2 節で述べた採点遷移モデルに基づき生成した。

表 1: 実データ 1 + 4.1.1 節の補完を用いた実験結果

	提案手法 1	提案手法 2	既存手法 1	既存手法 2
目的関数 (1)	48	48	762	772
目的関数 (2)	114.99	105.67	846.75	838.03

表 2: 実データ 2 + 4.1.1 節の補完を用いた実験結果

	提案手法 1	提案手法 2	既存手法 1	既存手法 2
目的関数 (1)	12	14	90	94
目的関数 (2)	18.91	19.92	98.21	100.70

4.3 比較手法

(A) を目的とした割り当て手法と、(A) と (B) を目的とした割り当て手法の各々に対して比較手法を用意する。どちらの比較手法も適応的な割り当てを行わず、採点が始まる前にすべての割り当てを決定する手法である（オフライン割り当てと呼ぶ）。また、どちらの手法においても講師から採点数を事前に指定されるものと仮定している（ただし、受講者は必ずしも指定された数だけ採点を行うとは限らない）。実データ 1 と偏在データについては、講師から指定された採点数を 3 とするのが最も自然だと考えられる。実データ 2 と均一データについては、講師から指定された採点数を判断するのは困難だが、本実験では他のデータと同様に講師から指定された採点数を 3 とする。なお、提案した適応的な割り当て手法は採点時間の有無で性能が変化するが、オフライン割り当て手法は採点時間の有無に影響を受けない。

4.3.1 ナイブなオフライン割り当て

(A) を目的とした割り当て手法に対応する比較手法である。全受講者の採点数と被採点数がともに 3 となるように、ランダムな割り当てを行う。

4.3.2 Longest Processing Time を応用したオフライン割り当て

(A) と (B) を目的とした割り当て手法に対応する比較手法である。既存研究 [12] は Longest Processing Time と呼ばれる手法を応用し、個々の答案に割り当てられる採点者の採点能力値の合計のばらつきがなるべく少なくなるような割り当て手法を提案している。この手法を利用し、目的 (B) を考慮したオフラインでの割り当てを行った。

4.4 実験結果

以下、3.1 節で提案した手法を提案手法 1、3.2 節で提案した手法を提案手法 2、4.3.1 節で説明した手法を既存手法 1、4.3.2 節で説明した手法を既存手法 2 と呼ぶ。また、目的関数 2 とアルゴリズム 2 で用いるパラメータ λ は 1 とした。

一つ目の実験では、実データ 1 と実データ 2 をもとに、4.1.1 節の方法で採点開始順序を生成した二種類のデータに対し、二つの提案手法と二つの比較手法を適用した。すべての場合について目的関数 (1) と目的関数 (2) の値を求めたところ、結果は表 1、表 2 のようになった。

目的関数 (1) と目的関数 (2) はともに値が小さいほど望ましいので、以上の結果は提案手法が既存手法よりも有効に働くこ

とを示している。詳しい考察は、続く実験で行う。

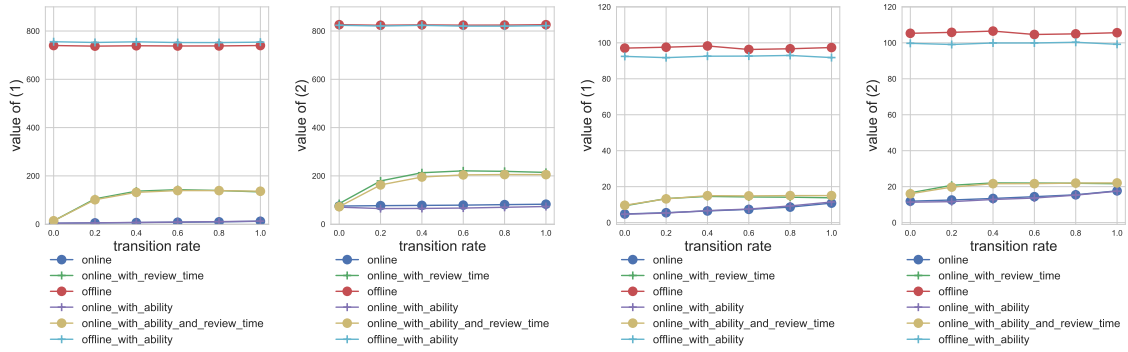
二つ目の実験では、実データ 1 と実データ 2 をもとに、確率 P が 0, 0.2, 0.4, 0.6, 0.8, 1 となるような採点遷移モデルに基づき採点開始順序を生成し、採点時間をすべて 0 とした場合と $[0, 10]$ において一様ランダムに生成した場合の計 12 種類のデータに対して二つの提案手法を適用した場合と、採点時間を考慮しない計 6 種類のデータに対して二つの既存手法を適用した。なお、既存手法に対して採点時間を考慮したデータを適用しない理由は、既存手法は採点時間を考慮したデータを用いても結果が変化しないからである。各手法について、各データを 100 個ずつ生成してシミュレーションし、目的関数 (1) の平均値と目的関数 (2) の平均値を求めた。

結果は図 4 のようになった。縦軸は目的関数の値、横軸は確率 P の値を表している。各図にプロットされた 6 種類の線は各々以下を表している。

- *online*: 採点時間を考慮しないデータに対して提案手法 1 を用いた際の目的関数の値（青色）
- *online_with_review_time*: 採点時間を考慮したデータに対して提案手法 1 を用いた際の目的関数の値（緑色）
- *offline*: 既存手法 1 を用いた際の目的関数の値（赤色）
- *online_with_ability*: 採点時間を考慮しないデータに対して提案手法 2 を用いた際の目的関数の値（紫色）
- *online_with_ability_and_review_time*: 採点時間を考慮したデータに対して提案手法 2 を用いた際の目的関数の値（黄色）
- *offline_with_ability*: 既存手法 2 を用いた際の目的関数の値（水色）

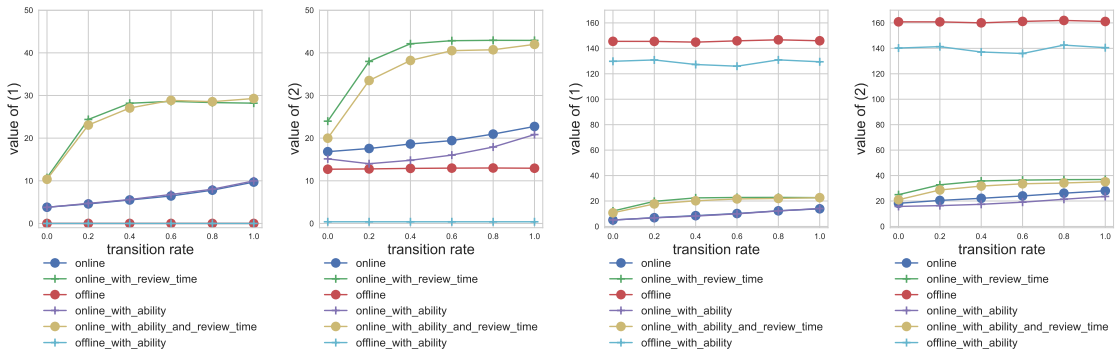
図 4 からは、いずれの場合も二つの提案手法が二つの既存手法を大きく上回っていることが読み取れる。また、採点時間を考慮する場合や、確率 P が高い（同じ採点者が連続する）場合に二つの提案手法が悪化することも読み取れる。提案手法 1 は目的関数 (1) の最小化を目標としたアルゴリズムで、提案手法 2 は目的関数 (2) の最小化を目標としたアルゴリズムであるが、図 4(a)(c)(d) では、二つの提案手法の性能差がほとんど観測されず、図 4(b) においてのみ性能差を観測できる。これらの結果はパラメータ λ に依存するため、一概に決めつけることはできないが、本実験においては、目的 (A) と目的 (B) を同時に目指す提案手法 2 は目的関数 (2) を小さくすると同時に、目的 (A) のみを目指す提案手法 1 と遜色なく目的関数 (1) の値を小さくすることができると考えられる。

三つ目の実験では、均一データと偏在データに対して、二つ目の実験と同様のシミュレーションを行った。結果は図 5 のようになった。図の表示形式についても図 4 と同様である。均一データでは実データと同じく提案手法が既存手法を大きく上回っているが、偏在データでは提案手法が既存手法を下回っている。これは偏在データでは受講者のドロップアウトが仮定さ



(a) 実データ 1・目的関数 (1) (b) 実データ 1・目的関数 (2) (c) 実データ 2・目的関数 (1) (d) 実データ 2・目的関数 (2)

図 4: 実データ + 4.1.2 節の補完を用いた実験結果



(a) 偏在データ・目的関数 (1) (b) 偏在データ・目的関数 (2) (c) 均一データ・目的関数 (1) (d) 均一データ・目的関数 (2)

図 5: 人工データ + 4.1.2 節の補完を用いた実験結果

れておらず、オフラインで割り当てられた答案がすべて採点され、目的関数 (1) が必ず 0 になるからだと考えられる。実際に図 5(a) から、既存手法 1 を用いた際の目的関数 (1) の値がすべて 0 となっていることが読み取れる。また、図 5(b) からは既存手法 2 を用いた際の目的関数 (2) の値が最も小さくなっていることが読み取れ、採点能力値を考慮したオフラインでの割り当てが有効に働いていることが分かる。ただし、実データ 1 のように多くの受講者の採点数が同じデータであっても、異なる採点数の受講者が存在する (図 (a) 参照) だけで途端に性能が悪化する (図 4(a)(b) 参照)。よって、オフラインでの割り当てが有効に働くのは一部の特殊な状況に限られると考えられる。

以上の実験から、多くの場合において提案手法は既存手法の性能を上回っていることが示唆された。

5. おわりに

本研究は以下の二点を目的とした採点者と答案の割り当て手法を提案した。

- (A) 各受講者の採点数と被採点数の差の絶対値の合計をなるべく小さくする
- (B) 各受講者の答案に割り当てられる採点者の採点能力値の平均をなるべく等しくする

また、提案手法の有用性を実データ・人工データを用いて確

認した。今後は提案したアルゴリズムの理論的な解析を行い、提案手法の頑健さを証明したい。また、採点能力値の推定や、各答案の採点結果を用いた最終的な評価値の推定などについても研究を行い、相互評価全体での有益な枠組みを提案したい。

文 献

- [1] Ittai Abraham, Omar Alonso, Vasilis Kandydas, and Aleksandrs Slivkins. Adaptive crowdsourcing algorithms for the bandit survey problem. In *Conference on learning theory*, pp. 882–910, 2013.
- [2] Enrique Sánchez Acosta, Juan José Escribano Otero, and Gabriela Christie Toletti. Peer review experiences for mooc. development and testing of a peer review system for a massive online course. *The New Educational Review*, Vol. 37, No. 3, pp. 66–79, 2014.
- [3] Dmytro Babik, Edward F Gehring, Jennifer Kidd, Ferry Pramudianto, and David Tinapple. Probing the landscape: Toward a systematic taxonomy of online peer assessment systems in education. In *Educational Data Mining (Workshops)*, 2016.
- [4] Stephen P Balfour. Assessing writing in moocs: Automated essay scoring and calibrated peer review (tm). *Research & Practice in Assessment*, Vol. 8, pp. 40–48, 2013.
- [5] Hou Pong Chan and Irwin King. Leveraging social connections to improve peer assessment in moocs. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 341–349, 2017.
- [6] Xi Chen, Qihang Lin, and Dengyong Zhou. Optimistic

- knowledge gradient policy for optimal budget allocation in crowdsourcing. In *International Conference on Machine Learning*, pp. 64–72, 2013.
- [7] Luca de Alfaro and Michael Shavlovsky. Crowdgrader: A tool for crowdsourcing the evaluation of homework assignments. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pp. 415–420, 2014.
- [8] Jorge Díez Peláez, Óscar Luaces Rodríguez, Amparo Alonso Betanzos, Alicia Troncoso, and Antonio Bahamonde Rionda. Peer assessment in moocs using preference learning via matrix factorization. In *NIPS Workshop on Data Driven Education*, 2013.
- [9] Erkan Er, Miguel Luis Bote-Lorenzo, Eduardo Gmez-Snchez, Yannis Dimitriadis, and Juan Ignacio Asensio-Prez. Predicting student participation in peer reviews in moocs. In *Proceedings of the Second European MOOCs Stakeholder Summit 2017*, 2017.
- [10] Iria Estévez-Ayres, Raquel M Crespo García, Jesús A Fisteus, and Carlos Delgado Kloos. An algorithm for peer review matching in massive courses for minimising students’ frustration. *J. UCS*, Vol. 19, No. 15, pp. 2173–2197, 2013.
- [11] Edward F Gehringer. A survey of methods for improving review quality. In *International Conference on Web-Based Learning*, pp. 92–97, 2014.
- [12] Yong Han, Wenjun Wu, and Yanjun Pu. Task assignment of peer grading in moocs. In *International Conference on Database Systems for Advanced Applications*, pp. 352–363, 2017.
- [13] David R Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, Vol. 62, No. 1, pp. 1–24, 2014.
- [14] Canvas Network. Canvas network courses, activities, and users (4/2014 - 9/2015) restricted dataset, 2016. <http://dx.doi.org/10.7910/DVN/XB2TLU>.
- [15] Daniel FO Onah, Jane Sinclair, and Russell Boyatt. Dropout rates of massive open online courses: behavioural patterns. *EDULEARN14 Proceedings*, pp. 5825–5834, 2014.
- [16] Laura Pappano. The year of the mooc. *The New York Times*, Vol. 2, No. 12, p. 2012, 2012.
- [17] Chris Piech, Jon Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. Tuned models of peer assessment in moocs. In *Educational Data Mining 2013*, 2013.
- [18] Lakshmi Ramachandran. *Automated assessment of reviews*. North Carolina State University, 2013.
- [19] Karthik Raman and Thorsten Joachims. Methods for ordinal peer grading. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1037–1046, 2014.
- [20] Nihar B Shah, Joseph K Bradley, Abhay Parekh, Martin Wainwright, and Kannan Ramchandran. A case for ordinal peer-evaluation in moocs. In *NIPS Workshop on Data Driven Education*, 2013.
- [21] Daniel S Weld, Eytan Adar, Lydia Chilton, Raphael Hoffmann, Eric Horvitz, Mitchell Koch, James Landay, Christopher H Lin, and Mausam Mausam. Personalized online education—a crowdsourcing challenge. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 1–31, 2012.
- [22] Yan Yan, Romer Rosales, Glenn Fung, and Jennifer G Dy. Active learning from crowds. In *International Conference on Machine Learning*, Vol. 11, pp. 1161–1168, 2011.