

# Crowd-in-the-Loopによる大規模学会のセッション作成の試み

小林 直樹<sup>†</sup> 松原 正樹<sup>††</sup> 田島 敬史<sup>†††</sup> 森嶋 厚行<sup>††</sup>

<sup>†</sup> 筑波大学情報学群情報メディア創成学類 〒305-8550 茨城県つくば市春日 1-2

<sup>††</sup> 筑波大学 知的コミュニティ基盤研究センター 〒305-8550 茨城県つくば市春日 1-2

<sup>†††</sup> 京都大学 京都大学大学院情報学研究所 〒606-8501 京都市左京区吉田本町 36-1

E-mail: †s1513102@u.tsukuba.ac.jp, ††{masaki,mori}@slis.tsukuba.ac.jp, †††tajima@i.kyoto-u.ac.jp

あらまし 大規模な学会では発表論文が数百件にも上ることがあり、それらの論文からセッションを作成することはプログラムチェアにとって大きな負担である。本研究では、群衆によるセッション作成の問題点を整理し、セッション作成のクラウドソーシング手法を提案する。提案手法は、投票タスクによる論文とセッション名の紐付けと、提案アルゴリズムによるタスク結果からセッション生成、そして改善タスクによる生成されたセッション案の修正の3つから構成される。提案手法を用いて大規模会議のセッション作成を実施した結果、88.5%の論文がシステムが提案したセッションに採択され、プログラム編成会議の負担が削減された。

キーワード クラウドソーシング, セッション作成, マイクロタスク, Human-in-the-Loop

## 1. はじめに

大規模学会のプログラム作成はセッションの作成とスケジューリングの二段階で行われる。セッションの作成では、全論文を共通のトピックを有するグループごとに分割する。セッションのスケジューリングでは、分割されたグループごとに日時と場所を割り当てる。例えば2017年のDEIMでは論文数が369件で、79個のセッションが3日間の日程にスケジュールされた。

大規模学会のプログラムを作成するに当たっては、次の4つの制約を考慮する必要がある。

(1) **Paper-level** ある論文はあるセッションに入れるべきである、もしくは入れるべきではない。

(2) **Session-level** あるセッションは特定の会場や日程に入れるべきである。

(3) **Intra-session-level** 同一セッションに同じ所属の発表者が半分以上いてはいけない。

(4) **Inter-session-level** 同じ時間に同じ発表者が複数のセッションに存在してはいけない。

(1), (3) は個々のセッション内容に関する制約であり, (2), (4) は会議全体のスケジュールに関する制約である。

これらの制約を考えながら、プログラムチェアが学会プログラムを作成するのは時間がかかり、大きな負担となる。そこで、本論文では、マイクロタスク型クラウドソーシング技術を利用して、プログラム作成を支援する仕組みを提案する。群衆を構成するものは、著者と、プログラム委員やコメンテータなどのエキスパートである。これらの数百人規模の人々が広く薄く負担する。

本論文では特に、学会プログラム作成のうち(1)(3)の制約を扱う問題に取り組む。すなわち、学会のプログラムを構成するセッションの集合を作る事が目的となる。

図1に、提案手法の全体像を示す。提案手法では、次の手順でセッション案を作成する。(1) 著者による希望セッションの

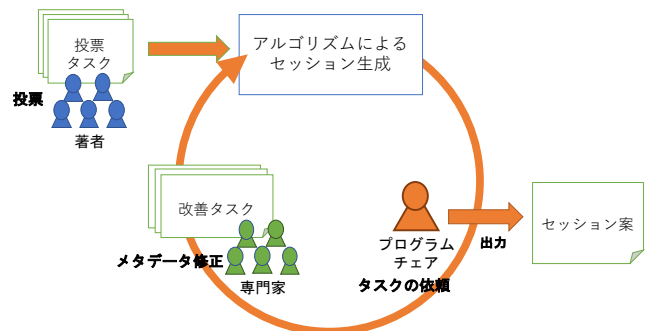


図1 提案手法の全体像

投票タスクを行なう。(2) アルゴリズムによって、セッション案を生成する。(3) プログラムチェアが、最終案にするか改善するかを選択する。最終案にする場合には出力して終了する。(4) 改善をする場合には、プログラムチェアが選んだセッション案を元に、エキスパート(プログラム委員など)による改善タスクを行なう。以上の(2)~(4)を、最終案ができるまで繰り返す。このように、アルゴリズムによる案の作成と、群衆による改善タスクを何度か繰り返すことによって、よりよいセッションプログラムの作成を目指す。

群衆による学会プログラムの作成は、次に示す5つの問題を考慮する必要があるため難しい。第一の問題は、良いセッションとは何かを決めなくてはならない。本論文では、より良いセッション名をもち、そのセッション名にふさわしく、かつ共通点の多い論文から構成されるセッションを、良いセッションと定義する。第二の問題は、背景知識が無ければ、より良いセッションを考えるのが難しいことである。後述するが、DEIMにおいては同じ論文キーワードを持つ論文の平均的な数は1.3個であり(図2)、キーワードを単純に用いたグルーピングは不可能である。また、キーワードの統合を行うことも難しい。例えば、機械学習と深層学習というキーワードが同一であるかどうか

かを判断するだけでも、専門知識が必要となる。第三の問題は、プログラムチェアの意向の反映である。プログラムチェアは作りたいセッションがあるが、その意向通りに著者が投票するとは限らない。第四に、群衆(特に著者)は必ずしも正しい入力を行うとは限らないということである。したがって、なんらかのクリーニングの作業は必要である。第五に、Paper-level 及び Intra-session-level の制約を満たしながら、全ての発表論文がセッションの枠に納まるようにしなければならないという問題がある。

本論文では、以上の問題を考慮した手法を提案し、実際の学会のプログラム作成に適用した結果を報告する。

論文の構成は次の通りである。2. 節では関連研究について説明する。3. 節では、セッション作成の問題と解決策について議論する。4. 節では、3. 節で述べた解決策について詳細に説明する。5. 節では、提案手法を用いた Crowd-in-the-Loop による DEIM2018 におけるセッション作成の試みについて説明する。6. 節はまとめと今後の課題である。

## 2. 関連研究

**群衆によるセッションの作成** 群衆が大規模学会でセッション作成を行う研究がいくつか存在する。Cobi [1] は、学会の参加者がセッションの制約を満たし、また、参加者の趣向に沿ったプログラムを作成するツールである。Cobi は、論文のクラスタリング、条件の取得、スケジューリングの3つのステップで大規模学会のセッション作成を行う。Frenzy [2] は、論文のカテゴリといったメタデータの抽出、抽出されたメタデータを用いて条件を満たすセッションの作成という2つのステップで大規模学会のセッション作成を行うツールである。2つ目のステップでは、同じ場所にいるコミュニケーションが可能な少数のボランティアが制約を満たすセッション作成を行う。

Cobi, Frenzy はどちらも群衆による協調的な作業を通してセッション作成を行う一方、本研究では、メタデータの獲得をマイクロタスクにより行い、アルゴリズムによりセッション提案を行う。ここでマイクロタスクは、群衆同士でのやり取りを行わない自己完結型のタスクを指す。マイクロタスクを用いることで、群衆への依頼が容易となる。

Frenzy のメタデータの獲得とメタデータを用いたセッションの作成の2つのステップに分割する手法は本研究でも同様である。Frenzy ではメタデータの獲得のために論文のカテゴリの抽出を行うが、本研究では、メタデータの獲得のために、論文のセッション名への希望調査を行う。

マイクロタスクによる会議プログラムの作成を行った先行研究 [3] があるが、キーワードと論文の関係が疎である場合、実際にセッションを作成することは難しい。図2は、DEIM2018のキーワードの各論文への出現回数を表した図であるが、ほとんどのキーワードは論文に1回しか出現していないため、先行研究をセッション作成に適用することは難しい。

また、会議のスケジュール管理システムとして広く使われているものに Confer [4] がある。Confer は参加者に聴講する発表論文を尋ねる。学会の主催者は、その情報を基に論文毎の人

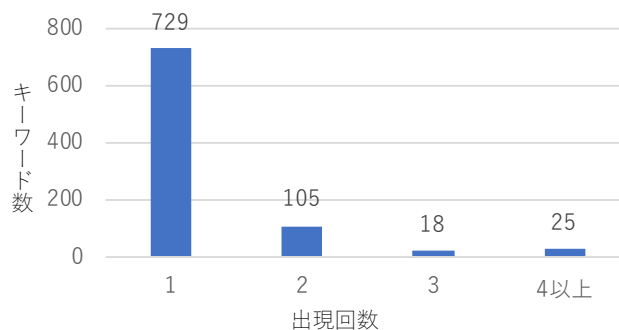


図2 DEIM2018におけるキーワードの論文への出現回数

気度を推定することが可能となる。Confer 自身はセッション作成機能をもたないが、Confer による論文の人気度はセッション作成の際の参考とすることができる。

**Human Computation** 本論文で提案する手法は、Human Computation [5] を用いて探索範囲を削減することで、計算機では困難な解の探索を可能にする点が特徴である。同様の手法が用いられているシステムとして、CrowdPlanr [6] がある。CrowdPlanr は、与えられた観光地の集合から Human Computation を用いて旅行計画を作成するシステムである。ある観光地について次にどの観光地に行くのが良いか複数人に質問して回答を集計し、Greedy に探索範囲を狭めて旅行計画を作成する。本論文でも、Greedy にセッション生成を行うが、アルゴリズムによるセッション生成結果に対して群衆が修正を行う点で異なる。

**群衆が関与するデータ分類** 群衆をデータ分類やクラスタリングに関与させる研究が行われている。例えば、クラウドソーシングはデータの分類に広く用いられる。典型的な方法は、クラウドソーシングによってラベルをデータ内の項目に割り当て、分類のための教師データを構築することである [7] [8]。また、ワーカが分類結果を評価することで分類器を改善させる研究 [9] や、分類のための特徴をクラウドソーシングにより獲得する研究 [10] が行われている。

セッション作成問題は、発表論文をセッションに分類するという点でデータ分類の問題に類似し、本研究でも群衆によってデータを獲得する。しかし、作成するセッションの枠数が与えられているため、分類器によるセッション作成は難しい。

**集合分割問題** 生成されたセッション案に対して評価が与えられるとき、セッション作成問題は、論文の集合を論文の部分集合であるセッションに分割する際の評価の最適解を求める集合分割問題である。集合分割問題は制約条件を満たす解を求めることも難しく、NP 困難な組合せ最適化問題の中でも特に計算困難な問題として知られている [11]。大規模学会では、発表論文が数百件にも及び、セッション案の候補は膨大であり厳密解を求めることは困難である。

## 3. セッションプログラム作成の課題と提案手法の概要

セッションとは、共通のトピックでまとめられた発表論文の

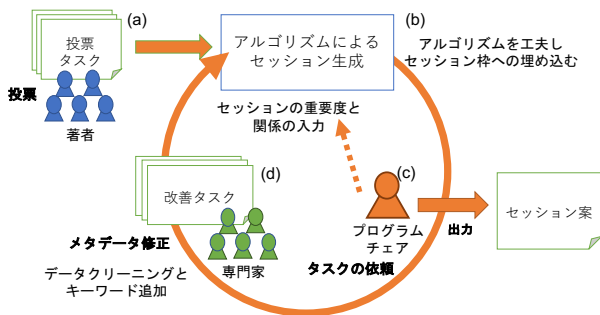


図3 提案手法の全体像の詳細

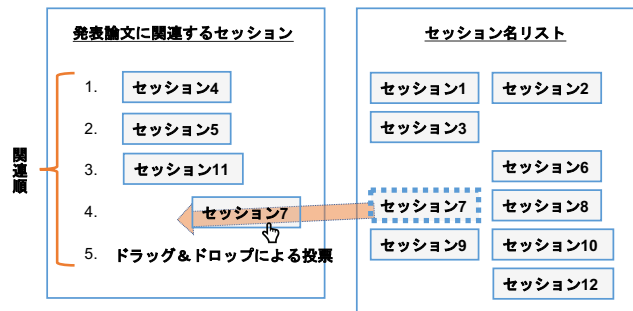


図4 投票タスクによる投票

部分集合と、共通トピックを表現するセッション名の組み合わせである。

1. 節で述べたとおり、群衆によるセッションプログラムの作成は、次の5つの課題がある。

**課題1** 良いセッションとは何かを決めなくてはならない。

**課題2** 背景知識が無ければ、より良いセッションを考えるのが難しい。論文のキーワードが疎なため、キーワードによるグルーピングは行えず、また、論文AとBを同じセッションに入れても良いかどうかは単純にキーワードを比較するだけでは困難である。

**課題3** プログラムチェアの意向を反映させる必要がある。プログラムチェアは作りたいセッションがあるが、その意向通りに著者が投票するとは限らない。

**課題4** 群衆(特に著者)は必ずしも正しい入力をするとは限らない。なんらかのクリーニングの作業が必要となる。

**課題5** 様々な制約を満たしながら、セッションの枠に納めなければならない。例えば、ある学会では、同一グループの論文は50%以下にしなければならないという制約がある。

### 3.1 良いセッションの定義

本論文では、より良いセッション名をもち、そのセッション名にふさわしく、かつ共通点の多い論文から構成されるセッションを、良いセッションと定義する。それぞれについて次に説明する。

**良いセッション名であること** 例えば、データベースの学会において、全てのセッション名を「データベース」とすれば、セッション名にふさわしい論文は全て揃うと考えられるが、それは、セッション名としてふさわしくはない。また、良いセッション名と思われるものであっても、同じセッション名があまりにも多く存在する場合、それは良いセッションとは言えない。このように、セッション名をどう選ぶかは、できたセッションの良さを計る重要な基準の一つである。

**含まれる論文がセッション名にふさわしいこと** セッション名にふさわしくない論文があるようなセッションは、良いセッションとは言えない。全ての論文がそのセッション名にふさわしい必要がある。

**含まれる論文に共通点が多いこと** 例えば、「情報検索(1)」「情報検索(2)」といった同じセッション名のセッションが2つ以上ある場合、それぞれのセッションにおいてはある程度共通点のある論文が揃っている方が望ましい。

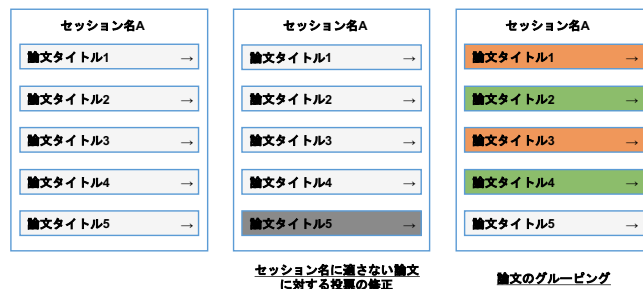


図5 エキスパートによる改善タスクの設計

### 3.2 提案手法概要

図3は、提案手法において群衆を用いたセッションプログラム作成の課題の対してどのようにアプローチするかを示したものである。

- 課題1に関しては、上記で定義したセッションの良さを反映できるような式を考案し、それを満たしたセッションを生成するよう、アルゴリズムを工夫する(図3(b))

- 課題2に対しては、論文につけられたキーワードだけでは不十分であることがわかっているので、著者に、セッション案に対して投票を行なってもらう(図3(a))。投票タスクの設計を図4に示す。この投票結果は、どのセッションに入れるかの参考にするだけでなく、アルゴリズムにおいて論文キーワードをベクトル化し、似たセッションに投票した論文のキーワードは近い関係にあるとみなす(図3(b))。さらに、エキスパートによる改善タスクにおいて、似た論文をグルーピングする作業を依頼し、その結果を新たなキーワードとみなして利用する(図3(d))。

- 課題3に関しては、プログラムチェアに、候補となるセッションの重要度を用意してもらい、それをアルゴリズムにおいて考慮する(図3(b))。また、アルゴリズムに必要ないくつかのパラメータを切替えて、プログラムチェアがより望ましいと思われるセッション案を選択する仕組みを用意する(図3(c))

- 課題4に関しては、エキスパートによる改善タスクにおいて、著者が投票したセッションに対するクリーニングを行なってもらう(図3(d))。図5に、エキスパートによる改善タスクの設計を示す。

- 課題5に関しては、各論文に対して、同じセッションで発表できそうな論文を見つける事が容易か否かの指標を用意し、

それを用いて Greedy にセッション枠に埋め込む (図 3 (b)). アルゴリズムによってセッション案を生成する際には, セッション名の優先度とセッションのクオリティ, そしてセッション作成の困難さの 3 つの指標のトレードオフとなる.

## 4. 提案手法

本節では, セッションを作成する提案手法について説明する. 本節の構成は次の通りである. 4.1 節では, 入出力の定義を行う. 4.2 節では, セッション投票タスクについて説明する. 4.3 節では, 論文キーワードのベクトル化について説明する. 4.4 節では, プログラムチェアが用意する情報, つまり各セッション名の重要度について説明する. 4.5 節では, 本アルゴリズムでセッションの良さを表す式を説明する. 4.6 節で, アルゴリズムの詳細を説明する. 4.7 節で, エキスパートによる改善タスクにおけるデータクリーニングとあらたなキーワード追加について説明する.

### 4.1 入出力

入力 セッション生成アルゴリズムの入力として以下の情報が与えられる.

- $P$ : 発表論文の集合
- $K$ : キーワードの集合
- $C$ : セッション名の集合
- セッション名の優先度  $I: C \rightarrow [0..1]$
- $C_{max}$ : 同一セッション名のセッション作成数の上限
- セッション作成枠  $F: \{n \mid f_{min} \leq n \leq f_{max}\} \rightarrow \mathbb{N}$
- $f_{min}, f_{max}$ : セッション作成枠の最大値と最小値

セッション名の優先度  $I$  は, セッション名の集合の中で優先して作成したいセッション名を優先して作成するための関数である. 優先して作成したいセッション名ほど 1 に近い値を返し, そうでないセッション名ほど 0 に近い値を返すように設定する. 同一セッション名のセッション作成数の上限  $C_{max}$  は, アルゴリズムの生成するセッション案で, 同一セッション名によって作成されるセッションの上限数である.

セッションの作成可能枠  $F$  は, セッションの各グループの論文数 (スロット数) に対する作成可能なグループ数を返す関数である.  $f_{min}, f_{max}$  は作成可能なセッションのスロット数の最小値と最大値を表している.  $x < f_{min}$  または  $x > f_{max}$  であるとき,  $F(x) = 0$  とする.

次に, 投票タスクの結果から, 以下の情報が与えられる.

- $B$ : 発表論文とセッション候補の投票関係  $B \subset P \times C$
- $W$ : 投票による重み  $W: P \times C \rightarrow [0..1]$

投票の重みは, 投票タスクによる投票の際に, 投票順位によって決定される 0 から 1 の係数である. 投票順位が高いほど投票の重みは 1 に近い.

また, 本アルゴリズムには 5 つのパラメータが存在する.

- $\alpha, \beta, \gamma$ : セッションのスコアを算出するときの重み係数
- $\delta, \epsilon$ : セッションのクオリティを算出するときの重み係数

出力 セッション  $s$  はセッション名  $c \in C$  と, 発表論文の部分集合であるグループ  $G \subset P$  の組み合わせ  $(G, c)$  で表し, アルゴリズムの出力としてセッションの集合  $S$  を返す.

セッション集合はセッションの作成可能枠を超えてはならない. セッション集合  $S$  から, スロット数が  $x$  であるセッションを全て取り出した集合を  $S_x$  と定義した場合,  $\forall x \in \mathbb{N}, |S_x| \leq F(x)$  となる.

### 4.2 セッション投票タスク

投票タスクはセッション名のリストから発表論文が発表できるセッション名を投票するタスクである. タスク実施の際, 発表論文に対し, 発表可能なセッションであるものを 1 つ以上, 出来るだけ多く投票してもらう.

投票タスクの設計を図 4 に示す. 著者が, 自身の発表論文に対して発表できると思うセッション名をセッションリストからドラッグアンドドロップにより投票する.

投票タスクの予備実験の結果より, できるだけ投票をしてもらうことが重要 [12] であることが分かったため, 質問文に投票の目標数を明示的に提示した.

投票タスクによる投票結果から, 投票の重み  $W$  を決定する. 投票の重みは, 論文の投票  $b$  に対して, その論文の総投票数を  $b_a$ , 投票順位を  $b_r$  としたとき,  $W$  を  $W(b) = (b_a + 1 - b_r)/b_a$  とする. これは, 投票順が 1 であるときに重みが 1 となり, 投票順が下がるたびに線形減少することを意味する.

投票タスクを発表論文に対して行うことで, 論文がセッション名に関連しているかどうかの関係を獲得できる. このタスク結果をもとにセッション案の生成を行う.

### 4.3 論文キーワードのベクトル化

論文で使用されるキーワードのベクトル化を行うために, 投票タスクの結果を用いる.

キーワードベクトルは, セッション名の要素をベクトル要素として持つ. キーワードに紐付く論文の集合に対して, 各セッション名の要素の投票結果の重みの総和がベクトルの値となる. つまり, キーワードベクトルを以下のように定義する. ここで,  $P(k)$  はキーワード  $k \in K$  を持つ論文の集合とする. また,  $c_i$  はセッション名の集合  $C$  の各要素である.

$$\vec{k} = \sum_{p \in P(k)} (W(p, c_1), W(p, c_2), \dots, W(p, c_{|C|}))$$

また論文のベクトルを論文が持つキーワードのベクトルの総和として表す.  $K(p)$  を論文が持つキーワードの集合とする.

$$\vec{p} = \sum_{k \in K(p)} \vec{k}$$

### 4.4 プログラムチェアによる入力情報

プログラムチェアは, アルゴリズムの入力として, セッション名の優先度を定義する.

セッション名の優先度は, セッション名に対してよりふさわしいと考えるセッションに対して優先度を高く設定する. 例えば, よりある分野に特化しているセッション名は, 一般的に優先度を高くするべきである. また, 学会が作りたいセッション名がある場合にはその優先度を高く設定する. アルゴリズムは, 優先度が高いセッション名のセッションを優先して作る.

#### 4.5 セッションの良さの定量化

セッション集合の列  $S = [s_1, s_2, \dots, s_{|S|}]$  が与えられた場合の、セッションの良さの定量化を行う。ここで、セッション  $s_i \in S$  は、セッション名  $c_i$  とセッションに含まれる論文集合  $G_i$  の組み合わせで表し、 $s_i = (G_i, c_i)$  である。

本論文では、セッションの良さ、つまりセッションの評価値をセッション集合の列に対するセッション名のプライオリティ  $P$  とセッションのクオリティ  $Q$  の総和で表す。本節では、この  $P$  と  $Q$  の定義を行う。

##### 4.5.1 セッション名のプライオリティ

セッション名  $c_i$  のプライオリティ  $P$  を以下の式で定義する。

$$P(c_i) = \max\{I(c_i) - \frac{\text{Count}(c_i)}{C_{max}}, 0\}$$

ここで、 $\text{Count}(c_i)$  は、セッション集合の列  $S$  の中でセッション名  $c_i$  が存在する数を表している。

セッション名の優先度  $I$  の値が高く、またセッション名の作成数が少ない場合にプライオリティ  $P$  は高くなり、その逆では低くなる。

##### 4.5.2 セッションのクオリティ

セッションのクオリティは、各論文の投票の重みの平均  $Q_B$  と、論文ベクトルの類似度の和  $Q_P$  によって表す。投票の重みは投票タスク及び改善タスクによって得られる論文の投票から得られる結果であり、論文ベクトルの類似度は論文に紐づくキーワードから得られる結果である。

$Q_B$  が高いことは、そのセッション名で強く発表を希望する論文が多いことを表し、一方で  $Q_P$  が高いことは、グループの論文のトピックが共通している可能性が高いことを表しており、どちらもセッションのクオリティに関わる。

投票の重みの平均  $Q_B$  は以下の式で表す。

$$Q_B(G_i, c_i) = \sum_{p \in G_i} \frac{W(p, c_i)}{|G_i|}$$

論文ベクトルの類似度の和  $Q_D$  は以下の式で表す。 $Q_D$  は、セッション内の論文集合の組み合わせに対するコサイン類似度の平均として定義する。

$$Q_P(G_i) = \sum_{p_i, p_j \in G_i, p_i \neq p_j} \frac{\cos(\vec{p}_i, \vec{p}_j)}{|G_i| C_2}$$

以上より、セッションのクオリティ  $Q$  を以下の式で定義する。ここで、 $\delta, \epsilon$  はそれぞれ、 $Q_B$  と  $Q_P$  に係る係数で、どちらを重視するかを決定する。

$$Q(G_i, c_i) = \frac{\delta Q_B(G_i, c_i) + \epsilon Q_P(G_i)}{\delta + \epsilon}$$

#### 4.6 セッション生成アルゴリズム

本節では、セッション生成アルゴリズムについて説明する。本アルゴリズムは、セッションの作成が最も難しい論文から、セッションのスコアが高くなるように Greedy にセッションを生成する。4.6.1 節で、セッションの作成の困難度を定式化し、4.6.2 節で、アルゴリズムで用いるセッションのスコアの定式

化を行う。ここで、セッションのスコアは、セッションの評価に加えて、セッション作成の困難度も指標とする。これは、セッションの評価の高いグループを選択するだけでなく、セッション作成ができない論文が最後に余るのを避けるためである。そして、4.6.3 節で、セッション生成アルゴリズムの説明を行う。

##### 4.6.1 セッション作成の困難さ

セッションの評価の高いグループを選択していただけでは、セッション作成ができない論文が最後に余ってしまうことがある。そこで、セッションの評価とは別に、各論文に対して、セッションの作成困難度  $D_P : P \rightarrow [0..1]$  を定義する。

セッションの作成が困難な論文とは、セッションへの投票数が少なく、また、投票したセッションが投票数の少ないセッションである論文である。このことからセッションの作成困難度  $D_P$  を以下のように定義する。

$$D_P(p) = \left| \bigcup_{c \in B_C(p)} B_P(c) \right|^{-1}$$

$p$  は発表論文を表し、 $B_C(p)$  は、発表論文が投票したセッションを表し、 $B_P(c)$  は、セッション候補  $c$  に投票した発表論文の集合を表す。つまりこの式は、発表論文が投票したセッションと同じセッションに投票した論文の数の逆数を表している。

また、セッションを作成するときに、残っている論文に対してセッション作成が容易な論文を多く残すことが、セッション作成ができない論文の数を減らす際に必要となる。ここで、グループの決定の際に用いるセッション困難度を以下のように定義する。

$$D(G) = \sum_{p \in P-G} 1 - D_P(p)$$

$D_P(p)$  を 1 から引くことで、セッション作成が容易な論文ほど値が高くなり、また、 $D : G \rightarrow [0..1]$  である。

##### 4.6.2 セッションのスコア

セッションのスコアを、セッションの評価で用いるセッション名のプライオリティとセッションのクオリティ、そしてセッション作成の困難度の 3 つの指標によって以下のように表す。

$$\text{Score}'(G, c) = \alpha D(G) + \beta P(c) + \gamma Q(G, c)$$

$$\text{ただし、}\alpha + \beta + \gamma = 1$$

$$\text{Score}(s) = \begin{cases} \text{Score}'(s) & (s \text{ satisfies constraints}) \\ 0 & (\text{otherwise}) \end{cases}$$

関数  $D, P, Q$  のうち、 $D$  と  $P$  はセッションの生成に応じて動的に値が変更される。セッションのスコアで、Intra-sesion-level の制約を満たしていない場合は 0 とする。ここで、パラメータ  $\alpha, \beta, \gamma$  は、0 から 1 までの間の実数値であり、 $\alpha + \beta + \gamma = 1$  とする。このスコアが高くなるよう、Greedy にセッションの生成を行う。

##### 4.6.3 アルゴリズム

投票タスクの結果からセッション案の生成を行うセッション

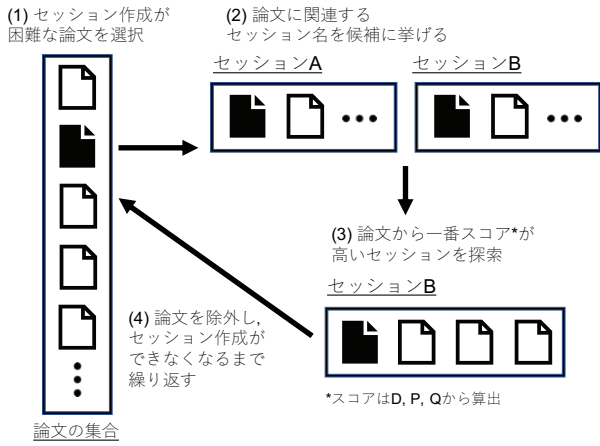


図6 セッション生成アルゴリズムのモデル図

### Algorithm 1 セッション生成アルゴリズム

Input:  $P, C, K, B, R$

Output:  $S$

```

1:  $S \leftarrow \phi$ 
2: while  $|P| \neq \phi$  do
3:    $p \leftarrow \arg \max_{p^* \in P} D_P(p^*)$ 
4:    $C^* \leftarrow \{c \mid c \in C, (p, c) \in B\}$ 
5:    $s \leftarrow \arg \max_{G^* \subset P, c \in C^*, G = \{p\} + G^*} \text{Score}(G, c)$ 
6:   if  $\text{Score}(s) > 0$  then
7:      $S \leftarrow S + \{s\}$ 
8:      $P \leftarrow P - G$ 
9:      $B \leftarrow B - \{b \mid c \in C, g \in G, (g, c) \in B\}$ 
10:  else
11:     $P \leftarrow P - \{p\}$ 
12:     $B \leftarrow B - \{b \mid c \in C, (p, c) \in B\}$ 
13:  end if
14: end while
15: return  $S$ 

```

生成アルゴリズムについて説明する。セッション案の生成は発表論文のグルーピングによって行われる。このアルゴリズムの手順を図6に示す。

セッション生成アルゴリズムは4つの手順で行われる。(1) 発表論文の中からセッションを作ることが最も困難な論文を選択する。(2) 選択した論文が投票したセッション名を挙げる。(3) 候補に投票された論文からグループを作成し、そのグループに対してスコア付けを行い、一番スコアの高いグループを決定する。そして、(4) グループの論文を除外して、発表論文がなくなる、もしくはグループを作成することができなくなるまでアルゴリズムを繰り返し、最終的に得られたグループの集合を生成されたセッション案とする。

セッション提案アルゴリズムを Algorithm 1 に示す。3行目では  $D_P$  が最も高い論文を選択する。4行目は、選択された論文が投票したセッション候補を得る。5行目では、セッション候補に対してセッションのスコアが最大となるセッション  $s$  を探索する。このとき、セッション内の論文集合は必ず4行目で選択された論文  $p$  が入るようにする。注意として、 $G$  は

$f_{min} \leq |G| \leq f_{max}$  であり、また、 $x = |G|$  として、 $F(x) > S_x$  となるようにする。セッションのスコアが0より大きい、つまり制約を満たすセッションが見つければ、7行目でセッションの追加を行う。また、セッションのスコアが大きい場合には、生成されたセッションに含まれる論文の部分集合を全体の論文集合から削除し、その論文に関連する投票関係も削除を行う。制約を満たすセッションが見つからない場合は、 $p$  の部分のみを削除する。この  $p$  はセッションが作れなかった論文を意味する。

#### 4.7 エキスパートによる改善タスク

アルゴリズムにより生成されたセッション案から、プログラムチェアによる修正が必要であると判断した場合に、エキスパートによるセッション案の改善タスクを実施する。改善タスクの設計を図5に示す。

改善タスクでは、セッション名に対して生成されたセッション案に含まれる論文集合を表示する。

エキスパートはセッション名の集合から自身の専門であるセッション名を選択し、そのセッション名に含まれる論文集合に対して2つの操作を行う。

- (1) 論文がセッション名に適さない場合、その論文を修正
- (2) 論文集合に対してグルーピング

(1)と(2)により投票の修正を行う。(1)で選択した論文に関して投票の修正を行う。論文タイトルの→をクリックすることでその論文の投票の修正を行う。投票の修正は、投票タスクと同じインターフェースで実施する。投票タスクには、その時点で投票されている情報が格納されており、そこから専門家は投票の修正を行う。(2)では、論文タイトルをクリックすることで色の変更が行われ、論文タイトルを同色にすることによりグルーピングを行う。グルーピングされた論文集合には、共通のキーワードを追加する。エキスパートによる投票の修正により、誤ったセッション名への投票の削除及び適切なセッション名の追加が行われ、また、同じグループと判定された論文は、同じキーワードが付与されることでグループ間の論文では論文ベクトルの類似度が高くなり、同じセッションに属する可能性が高くなる。

## 5. DEIM2018 における実践事例

本節では、DEIM2018で提案手法を適用した結果と、得られた知見について報告する。

### 5.1 入力データ

セッション作成の対象となる発表論文が286件、セッション名が65件、そしてキーワードの総数が879件であった。 $f_{min} = 5$ ,  $f_{max} = 6$ で、 $F(5) = 43$ ,  $F(6) = 13$ であった。プログラムチェアにより、セッション名の優先度が指定され、優先度が1のセッション名が15件、0.8が27件、0.6が20件、0.4が3件であった。

投票タスクは図4の設計を用いて、改善タスクは図5の設計を用いた。パラメータは、 $\delta = 5$ ,  $\epsilon = 1$ とし、投票結果を優先するようにした。また、 $\alpha, \beta, \gamma$ をステップ幅0.05ずつ実行した中で、論文の余りが最も少なく、その中でセッション評価が最も高いセッションを生成セッションとしている。

表1 投票タスク及び改善タスクの実施結果

タスク名	ワーカ	タスク実施回数	ワーカ数
投票タスク (1)	著者	286 件	286 人
投票タスク*	著者	63 件	63 人
改善タスク*	コメンテータ	59 件	53 人
改善タスク (1)	コメンテータ	44 件	40 人
改善タスク (2)	コメンテータ	9 件	6 人
改善タスク (3)	プログラム編集委員	6 件	4 人
改善タスク (4)	プログラム編集委員	9 件	6 人

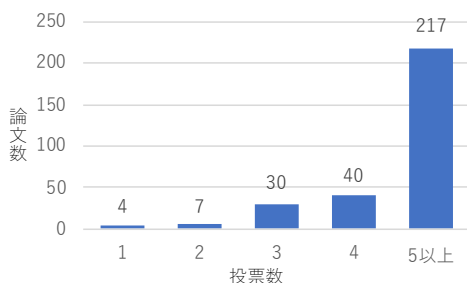


図7 投票タスク実施後の論文の投票数の分布

表2 プログラム編成会議におけるセッション編成作業の記録

実施年	作業人数	作業時間
2017	9 人	135 分
2018	4 人	70 分

## 5.2 実施結果

タスクの実施結果を表1に示す。投票タスクは発表者286名に依頼した。また、改善タスクはコメンテータ及びプログラム編集委員の計87名が行った。ここで、投票タスク(1)実施後、プログラムチェアがセッション名の追加を行った。投票タスク\*は、追加セッション名の中に、発表論文で発表できるものがあれば投票をしてもらうGoogle Formで作成した簡易的な投票タスクである。改善タスク\*は、間違っていると思う論文に対して投票の修正ではなく、そのセッション名への論文の投票の削除のみを行うタスクである。他のタスクは、投票タスクは図4、改善タスクは図5を用いている。

投票タスク実施後の発表論文の投票数の分布を図7に示す。多くの発表論文で、投票数が5以上と、多くのセッション名に投票されていることが分かる。

## 5.3 編成後セッションとの比較結果

本節では、提案手法で作成したセッション案に対する、編成会議でのプログラム委員による編成後のセッションとの比較結果について記述する。提案手法で作成したセッション案は、最後のタスク実施の終了時点でアルゴリズムにより生成したセッションであり、編成後セッションとの比較では、論文のセッション採択率によって行う。セッション採択率は、総論文数に対する同一セッションに採択された論文数の割合である。

提案手法が作成したセッション案の採択率は88.5%であった。これはつまり、286件の発表論文のうち、253件の論文が同じセッションとして採択され、33の論文が他セッションに移動したことを意味する。

表3 “機械学習”に対する類似度が高い上位5件のキーワード

キーワード
機械学習
クラスタリング
情報抽出
word2vec
ニューラルネットワーク
Text Mining

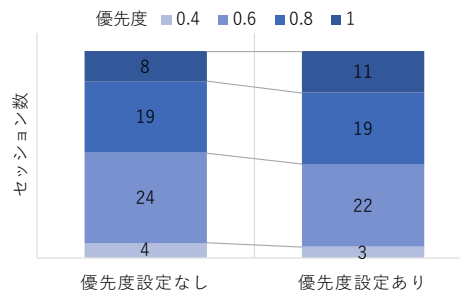


図8 各セッション名の優先度毎のアルゴリズムによる生成セッションの数

また、DEIMプログラム編成会議でのセッション編成時間の記録を表2に示す。注意として、DEIM2017においても、クラウドソーシングによるセッション作成の実施が試みられている。表2より、昨年と比較して、セッション編成作業の人数が少ないにもかかわらず、作業時間が昨年の半分程度となり、プログラム編成会議での作業負担が削減されたことが分かった。

## 5.4 キーワードのベクトル化

投票タスク及び改善タスクによって、キーワードのベクトル化が可能となる。表3はキーワード“機械学習”に対する、コサイン類似度が高い上位5件のキーワードである。“機械学習”に関連するキーワードが上位に現れていることが分かる。

ワーカによるタスクの実施によって、キーワードのベクトル化ができるようになり、数値的にキーワードの比較が可能となった。

## 5.5 セッション名の優先度の設定

セッション名の優先度を設定した場合のアルゴリズムにより生成したセッションと、優先度を設定しない場合の生成セッションを比較した図を図8に示す。優先度を設定しない場合は、全てのセッション名の優先度が1であるとすると、また、比較するセッション同士は同一のパラメータとする。

この図より、優先度を設定した場合の方が、設定しない場合と比較し、優先度が高いセッション名で作られたセッション数が多く、優先度が0.4、0.6と低いセッション名で作られたセッション数が少ないことが分かる。この結果から、優先度を設定することで、プログラムチェアの意向が反映されていることが読み取れる。

## 5.6 改善タスクの繰り返し

エキスパートによる改善タスク毎に生成されたセッションに対する編成後のセッションと比較した場合の論文採択率の変動を図9に示す。改善タスク(1)の採択率は44.4%、(2)では54.9%、(3)では59.0%、そして(4)では88.5%であった。エキ

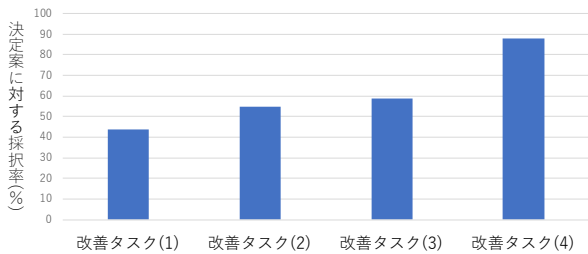


図9 各改善タスク実施後のアルゴリズムによる生成セッションと編成後セッションとの比較

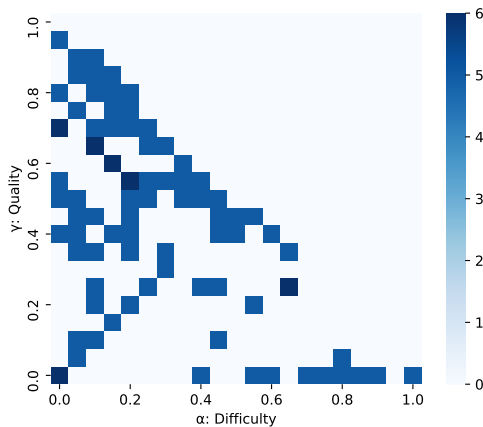


図10 各パラメータ毎のセッションを割り振れなかった論文数

スパートによる改善タスクの繰り返しによって、採択率が向上していることが分かる。

この結果から、エキスパートの改善タスクによって著者が行った投票タスクのクリーニングがなされ、アルゴリズムによって生成されるセッションの質が向上していることが考えられる。

### 5.7 パラメータの変更に対するセッション生成

図10はパラメータを0.05ずつ変更させた場合のアルゴリズムによってセッションが生成されなかった余りの論文数を表している。色の濃度が濃いほど余りの論文数が多いことを示している。最大の余りの論文数は6で、これは1つだけセッションが作れなかったことを意味し、多くのパラメータで、全てに論文にセッションを割り振ることができている。これは、論文への投票が充分に存在することを示している。また、論文のセッション作成の困難度に係るパラメータ $\alpha$ が大きい場合の方が余りの論文数が0となる場所が多く、論文のセッション作成の困難度が有効に働いていると考えられる。

## 6. まとめと今後の課題

本研究では、群衆によるセッション作成の課題を整理し、セッション作成を、アルゴリズムによるセッション案の生成と群衆によるセッション案の修正(Crowd-in-the-Loop)によって行う手法を提案した。提案手法は、投票タスクと改善タスクの2つのマイクロタスクと、定義したセッションのスコアによるGreedyなセッション生成アルゴリズムを用い、DEIM2018において提案手法によるセッション作成を実施した。

プログラム編成会議の編成結果より、88.5%の論文が提案手法で作成したセッションに採択され、編成会議でのセッション作成の作業負担が削減された。また、セッション結果に対する考察を行い、群衆によるセッション作成の課題に対して提案手法が有効であることを示した。

本研究では大規模学会のプログラム作成のうちセッション作成のみを扱ったが、群衆によるセッションのスケジュールも行うことが今後の課題として挙げられる。セッションのスケジュールも群衆が行うことで、プログラムチェアのプログラム作成の負担がさらに軽減できると我々は考えている。

## 謝 辞

本研究の一部はJST CREST (#JPMJCR16E3)の支援による。また、タスクの実施に協力していただいた方々にこの場を借りて感謝する。

## 文 献

- [1] Haoqi Zhang, Paul André, Lydia B. Chilton, Juho Kim, Steven P. Dow, Robert C. Miller, Wendy E. Mackay, and Michel Beaudouin-Lafon. *Cobi: communitysourcing large-scale conference scheduling*. In *CHI*, pp. 3011–3014, 2013.
- [2] Lydia B. Chilton, Juho Kim, Paul André, Felicia Cordeiro, James A. Landay, Daniel S. Weld, Steven P. Dow, Robert C. Miller, and Haoqi Zhang. *Frenzy: collaborative data organization for creating conference sessions*. In *CHI*, pp. 1255–1264, 2014.
- [3] Emi Sakurai, Atsuyuki Morishima, Kosetsu Ikeda, and Nobutaka Suzuki. *Bookshelf problem: A human-in-the-loop approach for data grouping without complete information*. In *iConference*, 2016.
- [4] Confer. <http://confer.csail.mit.edu>.
- [5] Edith Law and Luis von Ahn. *Input-agreement: a new mechanism for collecting data using human computation games*. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI*, pp. 1197–1206, 2009.
- [6] Ilia Lotosh, Tova Milo, and Slava Novgorodov. *Crowdplanr: Planning made easy with crowd*. In *29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013*, pp. 1344–1347, 2013.
- [7] Sudheendra Vijayanarasimhan and Kristen Grauman. *Large-scale live active learning: Training object detectors with crawled data and crowds*. In *CVPR*, pp. 1449–1456, 2011.
- [8] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. *AIDR: artificial intelligence for disaster response*. In *WWW'14*, pp. 159–162, 2014.
- [9] Chong Sun, Narasimhan Rampalli, Frank Yang, and AnHai Doan. *Chimera: Large-scale classification using machine learning, rules, and crowdsourcing*. *PVLDB*, Vol. 7, No. 13, pp. 1529–1540, 2014.
- [10] Jia Deng, Jonathan Krause, and Li Fei-Fei. *Fine-grained crowdsourcing for fine-grained recognition*. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2013.
- [11] 梅谷俊治. 大規模な集合分割問題に対する局所探索法(最適化の基礎理論と応用). 数理解析研究所講究録, Vol. 1879, pp. 91–96, apr 2014.
- [12] Naoki Kobayashi, Masaki Matsubara, Keishi Tajima, and Atsuyuki Morishima. *A crowd-in-the-loop approach for generating conference programs with microtasks*. In *2017 IEEE International Conference on Big Data (Big Data)*, pp. 4394–4396, Dec 2017.