Crowdsourcing with Diverse Groups of Users

Sara Cohen  Moran Yashinski
Team Formation problem

• Example: Forming an education board

• Required skills:
  • School Principal (SP)
  • High School teacher (HS)
  • Elementary School teacher (ES)
Team Formation problem with Communication Cost

- **Goal**: Find a team that has all required skills, while minimizing communication cost

- **Examples of communication costs**
  - Distance in the social network
  - (An inverse of) the number of papers each 2 experts published together
Research Question

• What if we wanted to define diversity based on the properties?
  - Gender, Income, Age, Religion, Location, etc.

• We would like to define target diversity function for the different experts’ properties

• **Goal:** Efficiently find a team that has all required skills, and is as close as possible to the desired target diversity
Team Formation with Target Diversity constraint

- Target Diversity based on **Properties**

- **Goal**: Efficiently find a team that has all required skills, and is as close as possible to the desired target diversity

- **Distribution Cost** = $|\text{Team Diversity} - \text{Target Diversity}|_1$

- **Example**:
  - Gender Target Diversity: $[\text{Male, Female}] = [\frac{1}{3}, \frac{2}{3}]$
  - Income Target Diversity: $[\text{High, Medium, Low}] = [\frac{1}{2}, \frac{1}{4}, \frac{1}{4}]$
What are we going to discuss?

• Research Question: diversity based on personal properties ✓
• Advantages of Diversity (or.. why is it interesting?)
• Related work
• Algorithms and computational considerations
  • Fixed Parameters Tractable (Optimal) Algorithm
  • Greedy Approximation Algorithm
• Experimental Results
• Conclusions
Advantages of Diversity (or.. why is it interesting?)

• Advantages in the workplace
  • Increase in productivity and creativity (innovative solutions)
  • Increase morale in workplaces
  • Positive reputation/attraction of quality human resources
• When crowdsourcing, it is important to consider different points of views
• Defining the diversity of a team
  • Program committees
  • Adopting affirmative actions
Related Work

• Team formation with Communication Cost
  • Goal: Find a team that has all required skills, while minimizing communication cost (e.g. Sum of Distances, Diameter)

• Diversity in terms of social influence
  • Depends on the social influences between candidates
  • Low social influence is correlated with high productivity

• Diversity in query answering
  • The goal is to maximize the diversity of the results
  • Diversity based on different criteria (e.g. content, novelty and coverage)
What have we achieved?

• Finding an optimal solution is NP-complete

• Naïve algorithm
  • Check all possible options and finds optimal solution
  • Time complexity: $O(|C||S||S||P|)$
  • Intractable in practice as $|C|$ might be huge

• Fixed Parameter Tractable (Optimal) Algorithm
  • Find an optimal solution in time complexity which is $poly(|C|)$ times $exp(|S|, |P|)$

• Greedy Approximation Algorithm
  • Time complexity: $poly(|S|, |C|)$
  • Guaranteed to return 1/2-approximation of the optimal solution
Fixed Parameter Tractable (Optimal) Algorithm

• Finds optimal solution
• Complexity time: $poly(|C|)$ times $exp(|S|, |P|)$
• Using preprocessed data structures in order to improve runtime performance
• Use the notion of Abstract (Optimal) Templates and Concrete Templates
Abstract (Optimal) Templates, Concrete Templates: Example

• One property (Gender):
  • \([\text{Male, Female}] = \left[ \frac{2}{3}, \frac{1}{3} \right] \)

• \(S = \{SP, HS, ES\}\)

• Abstract Optimal Template
  • Achieves \textit{minimum distribution cost}
  • There could be many Abstract Optimal Templates

• Abstract Template (non optimal)

• Concrete Templates:
  • \(\text{gender}(SP) = F, \text{gender}(HS) = M, \text{gender}(ES) = M\)
  • \(\text{gender}(SP) = M, \text{gender}(HS) = F, \text{gender}(ES) = M\)
  • \(\text{gender}(SP) = M, \text{gender}(HS) = M, \text{gender}(ES) = F\)
FPT Optimal Algorithm: Data structures

• Used to optimize runtime performance
• Hashset $\mathbb{H}$ to hold all the abstract templates
  • To avoid evaluating an abstract template more than once (very costly)
• minHeap $\mathbb{M}$ to efficiently return the abstract template which has minimum cost
• Structure $\mathcal{SPC}$
  • Calculated offline

Skills

Properties

Candidates
Calculate Optimal Abstract Templates and insert to $\mathcal{H}$ and $\mathcal{M}$

Extract Abstract Template A from $\mathcal{M}$

Check in $\mathcal{S}\mathcal{P}$ for candidates which satisfy all concrete templates (for all properties)

Create NEXT Abstract Templates from A

Create Concrete Templates from A

If found, STOP and return

If not in $\mathcal{H}$, insert to $\mathcal{H}$ and $\mathcal{M}$
Greedy Approximation Algorithm

• Time complexity: \( poly(|S|, |C|) \)
• Using sets of candidates per skill
• Greedy solution: in each step chooses an unchosen skill and candidate with that skill which (locally) minimizes the distribution cost
Greedy Approximation Algorithm (cont.)

• Optimizing a function call \textit{benefit}, that is inversely proportional to the \textit{distribution cost}

• The \textit{benefit} function is a monotonic submodular function and therefore guaranteed to return $1/2$-approximation of the optimal solution
Experimentation

• Tested scalability as a function of $|C|$, $|S|$, $|P|$ and Property Range
  • Default values: $|S| = 8$, $|P| = 5$, $|C| = 100K$, Property Range = 4

• Types of synthetic datasets:
  • TC1 (random assignment)
    • Property values: assigned randomly using uniform distribution
    • Skills per candidate: randomly choosing between 1 and $|S|$ skills per candidate
  • TC2 (random assignment with 1 skill)
    • Property values: assigned randomly using uniform distribution
    • Skills per candidate: each candidate is given 1 random skill
  • TC3 (skewed distribution with 2 skills)
    • Property values and skills (2 skills per candidate) are assigned using a skewed distribution
Experimentation: Varying number of skills
Experimentation: Varying number of properties

![Graph showing the relationship between varying number of properties and performance metrics for different algorithms. The x-axis represents the number of properties (3, 5, 7), and the y-axis represents the metric values (log scale). The graph compares TC1FPT, TC2FPT, TC3FPT, TC1Greedy, TC2Greedy, and TC3Greedy algorithms. Each line represents a different algorithm, with distinct colors for easy identification. The metric values increase as the number of properties increases, indicating a positive correlation.]
Experimentation: Varying number of candidates
Experimentation: Varying property range
## Experimentation: Quality of Results (Greedy Vs. FPT)

<table>
<thead>
<tr>
<th></th>
<th>TC1</th>
<th>TC2</th>
<th>TC3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max diff</strong></td>
<td>0</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td><strong>Average over all test cases</strong></td>
<td>0</td>
<td>0.01</td>
<td>0.11</td>
</tr>
<tr>
<td><strong>Average over test cases in which greedy didn’t return optimal result</strong></td>
<td>0</td>
<td>0.25</td>
<td>0.29</td>
</tr>
</tbody>
</table>
Conclusions

• FPT Optimal Algorithm
  • Always returns an optimal result
  • Time increases exponentially with the number of skills, properties and property range
  • Increasing the number of candidates doesn’t impact running time (except when the data is skewed)
  • Might take long time to find the optimal solution (especially when the data is skewed)
  • Outperforms the Greedy Algorithm when there is little skew in the data

• Greedy Approximation Algorithm
  • Performs well under all types of data
  • Returns results close to optimal
Questions?