Leximax Approximations and Representative Cohort Selection

Monika Henzinger[1] Charlotte Peale[2] Omer Reingold[2] Judy Hanwen Shen[2]

[1]University of Vienna
[2]Stanford University

Stanford
Computer Science

Motivation

Finding Representative Cohorts

In settings like consumer panels, surveys, and civic participation committees, we wish to have a diverse or representative cohort to express the interests of the overall population.

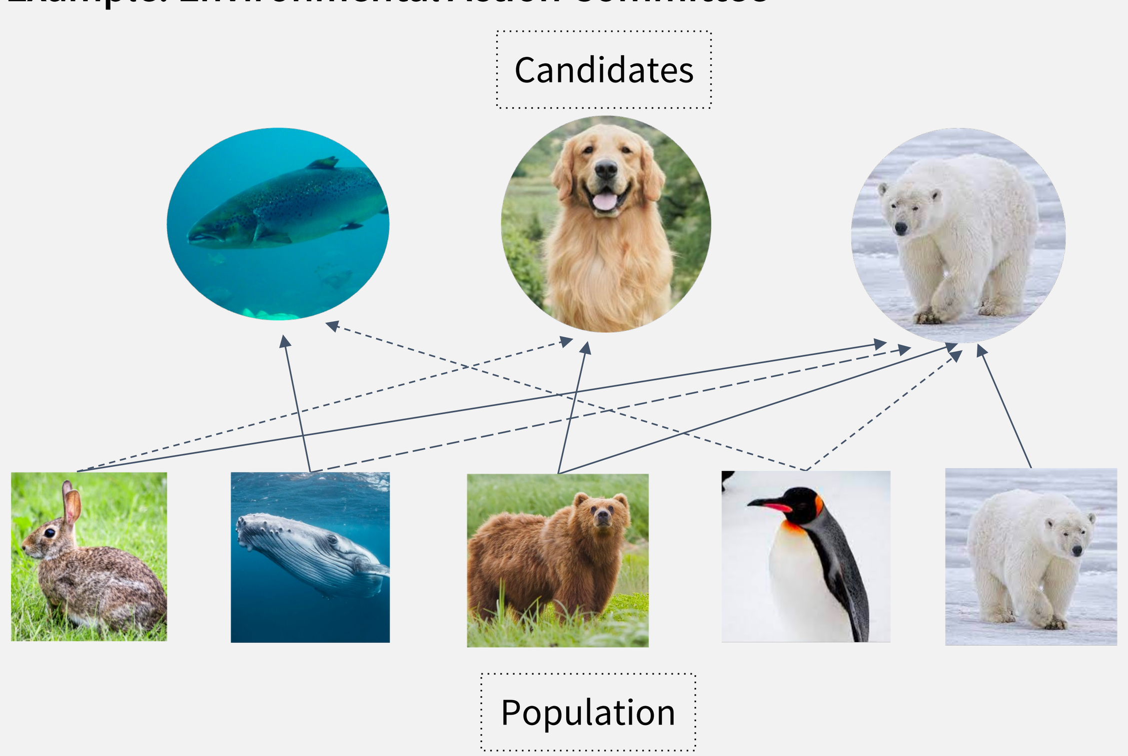
One notion of diversity

"we want a representative from every subgroup in the population"

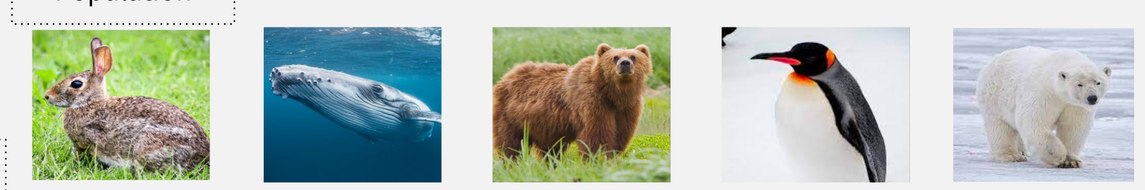
Another notion of diversity

"we want **good representation** of **every subgroup** in the population"

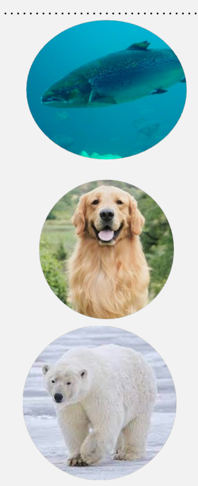
Example: Environmental Action Committee



Population



Candidates



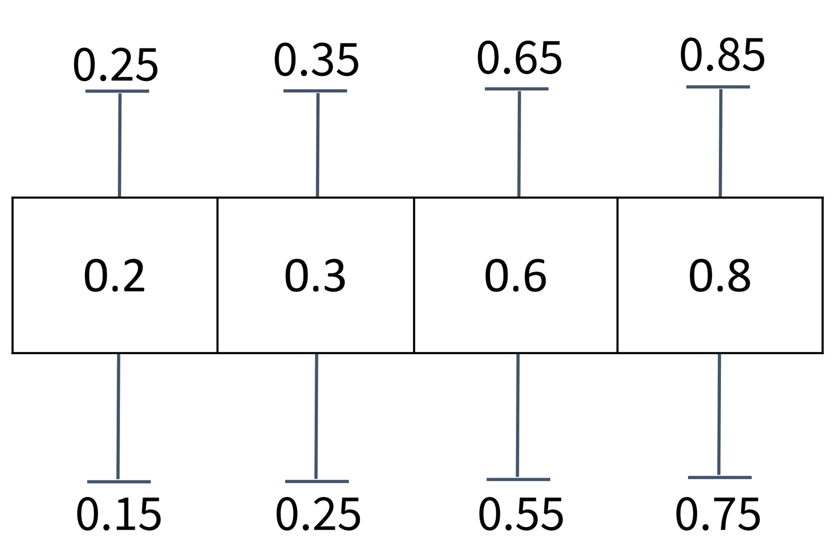
0.1	0.2	0.01	0.1	0.1
0.4	0.02	0.1	0	0.1
0.2	0	0.9	0.7	1.0

Approximate Leximax Definitions

*The leximax objective finds the best-possible guarantee for **all groups**, not just the worst-off. [1, 2]*
How can we approximate the leximax objective to be more robust to small fluctuations of worst group utility?

Element-Wise Approximation

▶ **Definition 5** (Element-wise leximax approximation). *Given a set of m groups $\mathcal{G}$ and set of potential solutions $\mathcal{S}$, let ℓ be the sorted vector of utilities attained by any lexima solution. We say that a solution $S \in \mathcal{S}$ is an α-element-wise leximax approximation if $\max_{i\in[m]}\{\ell_i - u(S, G_{[i]})\} \leq \alpha$.*



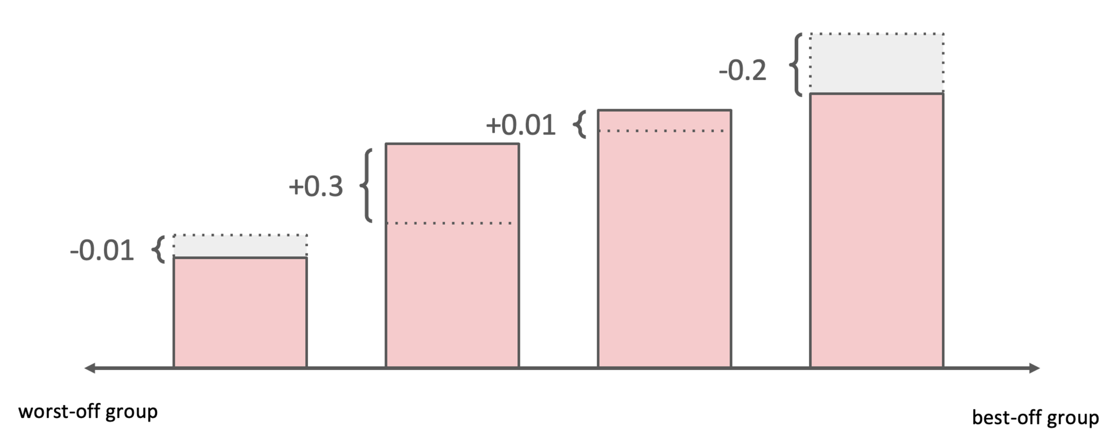
Easy to verify, but the requirement can be too strict and the solution might be difficult to find.

Recursive Approximation (stronger form of [3])

▶ **Definition 9** (ϵ-recursive leximax). *Given a set of m groups, $\mathcal{G}$, a set of potential solutions $\mathcal{S}$, and a choice of allowable 'slack' $\vec{\alpha} = (\alpha_1, ..., \alpha_m)$ with $\alpha_i \in \mathbb{R}_{\geq 0}$, recursively define the sets of solutions $\mathcal{S}_0^\alpha, ..., \mathcal{S}_m^\alpha \subseteq \mathcal{S}$ such that $\mathcal{S}_0^\alpha := \mathcal{S}$ and for each $i = 1, ..., m$,*

$$\mathcal{S}_i^\alpha = \{S \in \mathcal{S}_{i-1}^\alpha : u(S, G_{[i]}) \geq \max_{S' \in \mathcal{S}_{i-1}^\alpha} u(S', G_{[i]}) - \alpha_i\}$$

We say that $S \in \mathcal{S}$ is an ϵ-recursively approximate leximax solution if there exists an $\vec{\alpha}$ with $\max_{i\in[m]} \alpha_i \leq \epsilon$ such that $S \in \mathcal{S}_m^\alpha$.



This recursive definition allows us to consider small estimation errors recursively at each step.

Trade-off Approximation

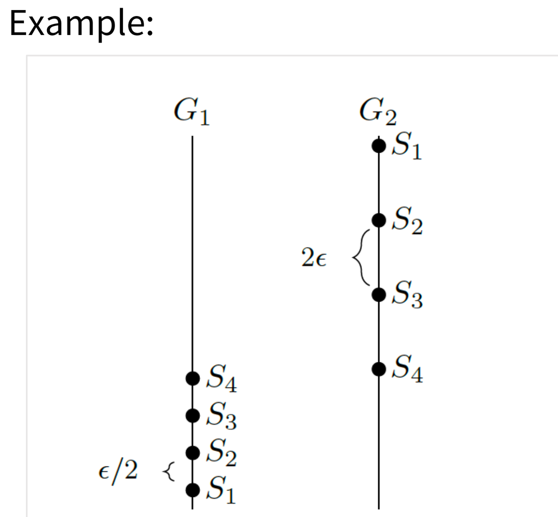
Definition (ϵ-tradeoff leximax)

Given a set of m groups, $\mathcal{G}$, and a set of potential solutions, $\mathcal{S}$, a solution $S \in \mathcal{S}$ is ϵ-tradeoff leximax if for any S' and i such that $u(S, G_{[i]}) < u(S', G_{[i]}) - \epsilon$, there exists a $j < i$ such that $u(S, G_{[j]}) > u(S', G_{[j]})$.

This definition guarantees that if we can find some other solution that does a lot better on some particular group, then this new solution must also decrease the utility of some worse-off group.

This definition is always guaranteed to exist while other relaxations might not always have a solution!

Example:



If we slightly modified the above definition to be:
$u(S, G_{[j]}) > u(S, G_{[j]}) + \epsilon$

Then there are no solutions in this example that satisfy this approximation.

Significant Leximax Approximation

Definition (ϵ-significant recursive leximax)

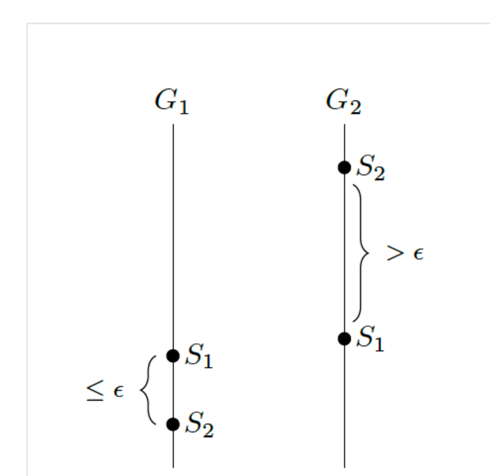
Given a set of groups $\mathcal{G}$ with $|\mathcal{G}| = m$ and a set of potential solutions $\mathcal{S}$, recursively define the sets of solutions $\mathcal{S}_0^\epsilon, ..., \mathcal{S}_m^\epsilon \subseteq \mathcal{S}$ such that $\mathcal{S}_0^\epsilon := \mathcal{S}$ and for each $i = 1, ..., m$,

$$\mathcal{S}_i^\epsilon = \{S \in \mathcal{S}_{i-1}^\epsilon : u(S, G_{[i]}) \geq \max_{S' \in \mathcal{S}_{i-1}^\epsilon} u(S', G_{[i]}) - \epsilon\}$$

We say that $S \in \mathcal{S}$ is ϵ-significant recursive leximax if $S \in \mathcal{S}_m^\epsilon$.

This definition allows us to constrain the choice of slack so we only consider solutions that significantly improve the quality of solutions.

Example:



S_2 improves G_2 utility significantly if we consider some slack for G_1

Thus improving the quality of the final solution

Finding Leximax-Optimal Solutions

γ_m : min utility across m groups
x_i : decision variable for candidate i
v_{ij} : utility of j-th representative for the i-th candidate

Solutions via Linear Programming

$$\begin{aligned} \text{maximize}_{x,\gamma_m} \quad & \gamma_m \\ \text{subject to} \quad & \textstyle\sum_{i=1}^n x_i = k \\ & 0 \leq x_i \leq 1 \\ & \textstyle\sum_{i=1}^n \sum_{G_j \in S} v_{ij} x_i \geq \sum_{s=1}^l \gamma_s^* \\ & \forall l = 1, \ldots, m, \forall S \subseteq \mathcal{G} \ s.t. \ |S| = l \end{aligned}$$

M linear programs

Fractional relaxation of cohort membership

Exponential number of constraints

We can solve this in polynomial time using the ellipsoid method with a separation oracle. (Lemma 16)

Integer program problem (hardness results)

$$\begin{aligned} \text{maximize} \quad & \gamma \\ \text{subject to} \quad & \textstyle\sum_{i=1}^n x_i = k \\ & x_i \in \{0, 1\} \\ & \textstyle\sum_{i=1}^n x_i v_{ij} \geq \gamma \quad \forall j = 1, \ldots, m \end{aligned}$$

The maxmin version of the integer cohort selection problem with linear utilities is NP-Hard (Lemma 20)

Discussion & Applications

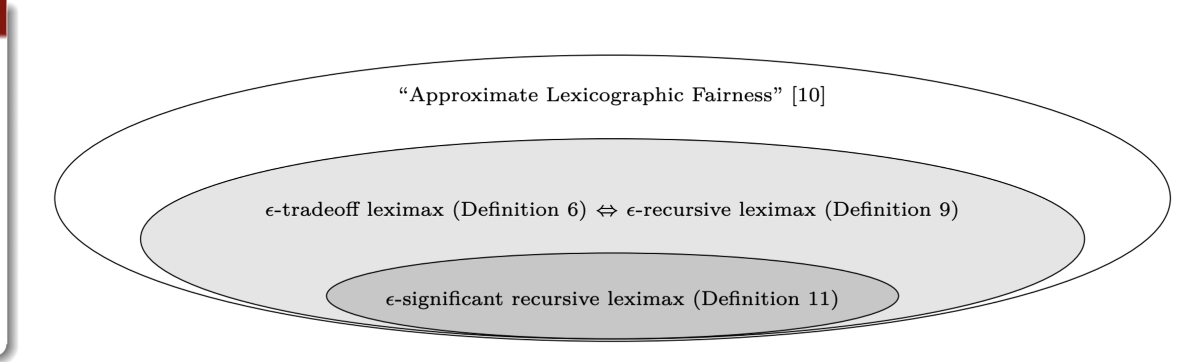
Main Takeaways

- Representation from the bottom up requires trade-offs between subgroup/individual utilities
- One notion to achieve the best possible outcome for all groups starting with the worst group is to find lexicographically maximal utilities
- To find robust leximax solutions, we can turn to approximate definitions of lexicographical maximality

Extensions and future work

- Develop approximate leximax notions based on multiplicative errors
- Solve or formulate cohort selection problems with non-linear utilities
- Model potential strategic manipulation of subgroup utilities for representative cohort selection
- Applications to dataset curation: how do we choose a representative test set for an underlying population without relying on discrete features?

How are these definitions related?



For an algorithmic approach for finding a tradeoff approximation, one option is to use an recursive approach. In fact, we prove the two definitions are equivalent!

The general form of recursive approximation proposed in [3] is less strict, while the significant leximax approximation definition we propose is more strict.

References

[1] *Jon Kleinberg, Yuval Rabani, and Éva Tardos. Fairness in routing and load balancing. In 40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*

[2] *David Kurokawa, Ariel D. Procaccia, and Nisarg Shah. Leximin Allocations in the Real World. ACM Transactions on Economics and Computation, 6(3-4):11:1–11:24, October 2018.*

[3] Emily Diana, Wesley Gill, Ira Globus-Harris, Michael Kearns, Aaron Roth, and Saeed Sharifi-Malvajerdi. Lexicographically Fair Learning: Algorithms and Generalization.

Stanford University