Individually Fair Diversity Maximization

Ruien Li, Yanhao Wang
School of Data Science and Engineering
East China Normal University

NEURAL INFORMATION PROCESSING SYSTEMS

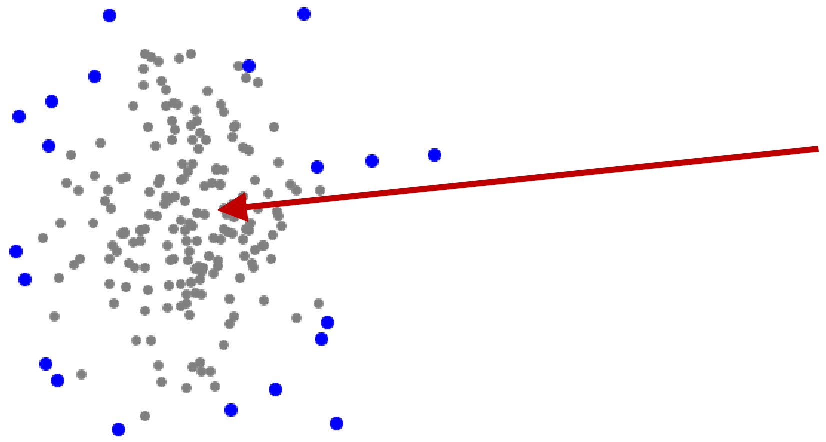
Motivations – Diversity Maximization

- Diversity maximization is a type of data selection. It extracts *k* diverse points from *n* points, key for apps like search, recommendations, feature selection.
- There are many different definitions on how to measure *diversity* of a certain subset.

1. **Max-Min Diversification:** $\mathsf{div}_{\mathrm{mm}}(S, P) = \min_{u \in S} d(u, S \setminus u)$.
2. **Max-Sum Diversification:** $\mathsf{div}_{\mathrm{ms}}(S, P) = \sum_{u \in S} \sum_{v \in S} d(u, v)$.
3. **Sum-Min Diversification:** $\mathsf{div}_{\mathrm{sm}}(S, P) = \sum_{u \in S} d(u, S \setminus u)$.

Motivations – Example

- Negative social consequences caused by Machine Learning algorithms.
- Max-Sum Diversification with no constraints tends to pick “outliers” and may include highly similar items in the solution – **unfair**!



"Who’s gonna advocate for me, huh?!"

[1] Wang, Y., Mathioudakis, M., Li, J., & Fabbri, F. (2023). Max-min diversification with fairness constraints: Exact and approximation algorithms. In Proceedings of the 2023 SIAM International Conference on Data Mining (SDM) (pp. 91-99). Society for Industrial and Applied Mathematics.

Definitions – Fairness

- Group Fairness / Partition Matroid:

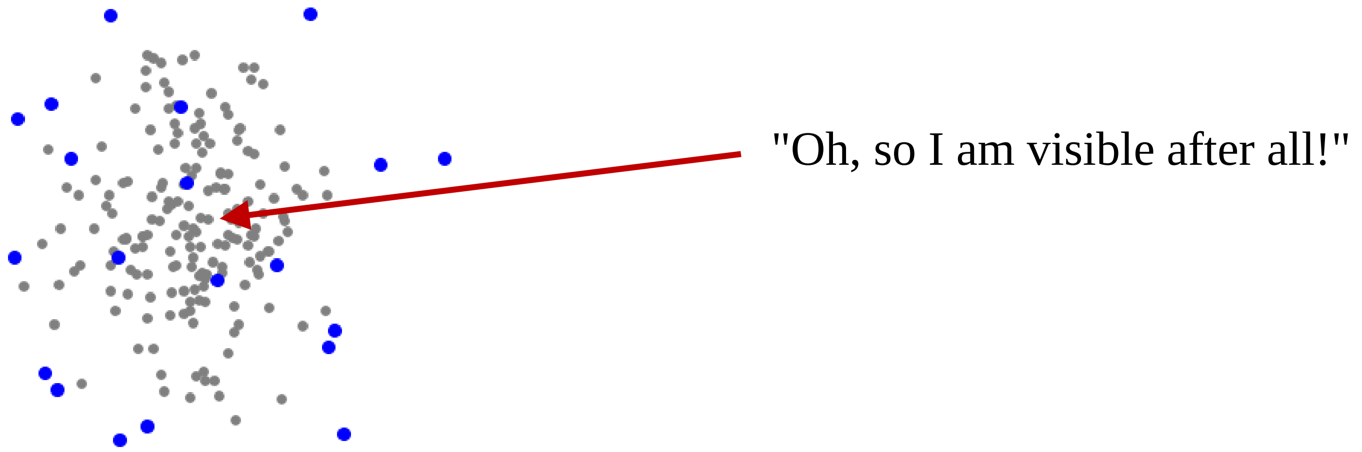
Each point in P denotes an individual associated with a particular demographic attribute, e.g., gender or race.

P is divided into different demographic groups, and an algorithm is required to select a subset S from P not only to maximize the diversity of S but also to ensure the selection of a pre-determined number of points from every group in S for equitable representation.

Definitions – Fairness

- Individual Fairness:

A data selection of a given point set P is fair if every point in P has a center among its $(|P|/k)$-closest neighbors.

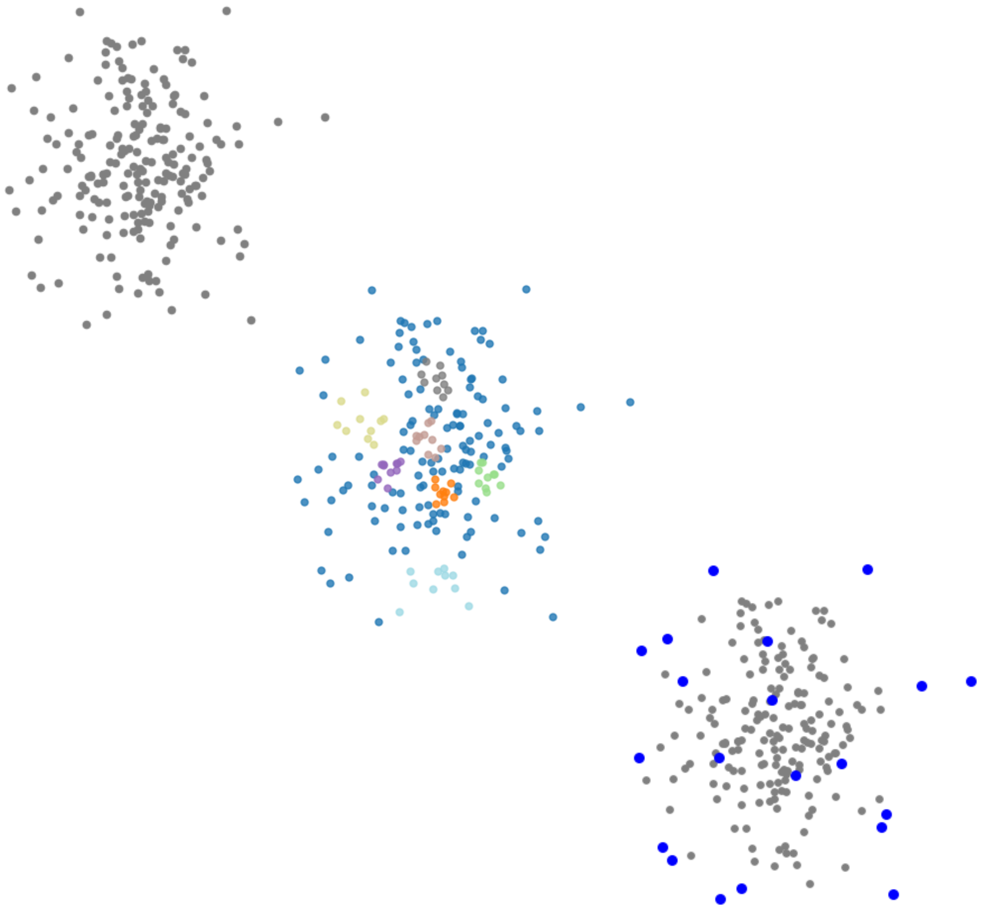


Contributions – Theory

- We propose ($O(1)$, 3)-bicriteria approximation algorithms for the individually fair variants of the three most common diversity maximization problems, namely, max-min diversification, max-sum diversification, and sum-min diversification.
- Specifically, the proposed algorithms provide a set of points where every point in the dataset finds a point within a distance at most 3 times its distance to its n/k -nearest neighbor while achieving a diversity value at most $O(1)$ times lower than the optimal solution.

Contributions – General Methods

- Step 1: Obtain *Individual Fairness Region* through our algorithms from the original dataset P.
- Step 2: Treat *Individual Fairness Region* as a type of "demographic attribute".
- Step 3: Implement the algorithms for diversity maximization under partition matroid (group) constraints.
- The results generated satisfy $(O(1), 3)$-bicriteria approximation!



Contributions – Experiments

- Numerical experiments on real world and synthetic datasets demonstrate that the proposed algorithms generate individually fairer solutions than unfair ones and incur only a modest utility loss for diversity maximization.

Contributions – Experiments

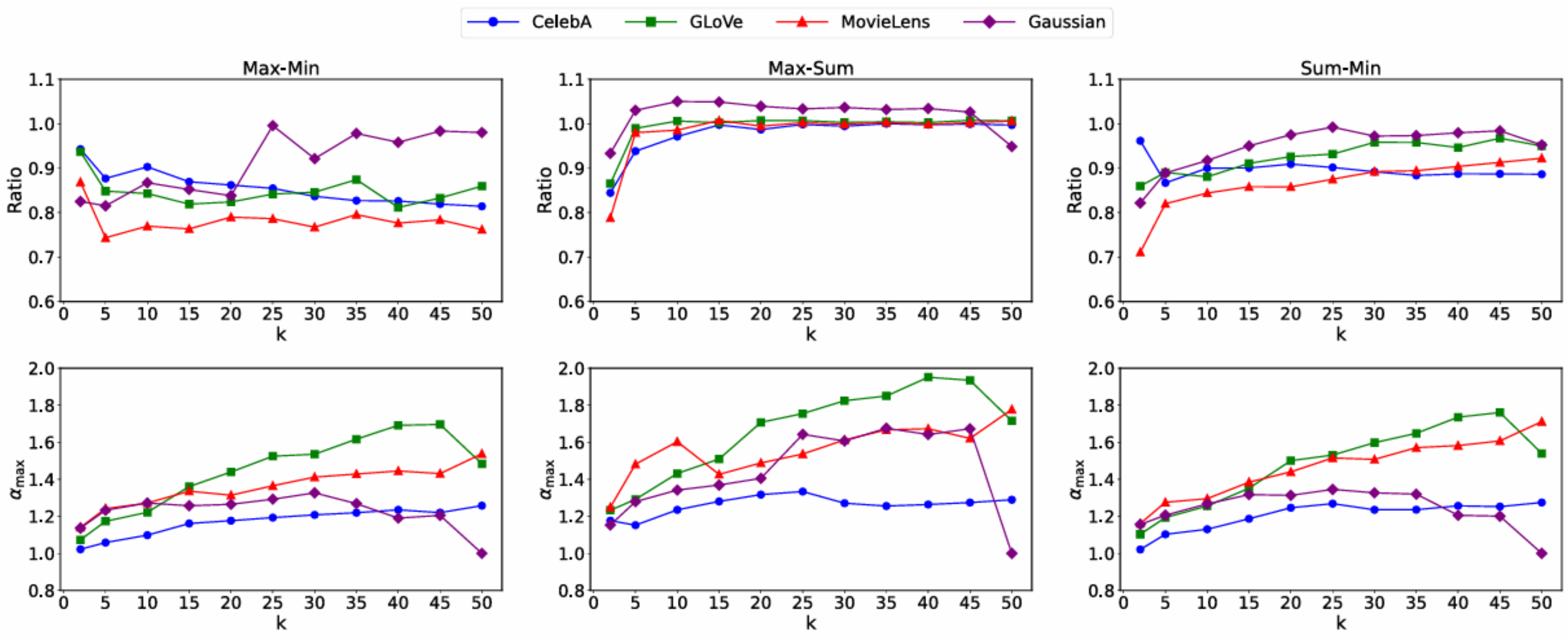


Figure 1: Overall experimental results. The first row illustrates the ratios of the diversity values of the solutions returned by our algorithms over the diversity values of the best solutions returned by unconstrained algorithms when $k = 2, 5, 10, \ldots, 50$ and $\alpha = 1$; the second row presents the $\alpha_{\max}$ values of the solutions returned by our algorithms.

Contributions – Experiments

Table 2: Ratios of the diversity values of the solutions of our proposed algorithms over the diversity values of the optimal unconstrained solutions by ILPs when $k = 5, 10, 20$ and $\alpha = 1$.

Dataset	**Max-Min**			**Max-Sum**			**Sum-Min**		
	$k = 5$	$k = 10$	$k = 20$	$k = 5$	$k = 10$	$k = 20$	$k = 5$	$k = 10$	$k = 20$
CelebA	0.827	0.807	0.794	0.923	0.967	0.985	0.703	0.769	0.842
GloVe	0.780	0.793	0.824	0.947	0.980	0.991	0.754	0.835	0.910
MovieLens	0.701	0.724	0.764	0.925	0.961	0.977	0.676	0.757	0.830
Gaussian	0.819	0.845	0.819	0.968	0.988	0.996	0.831	0.893	0.955

Contributions – Experiments

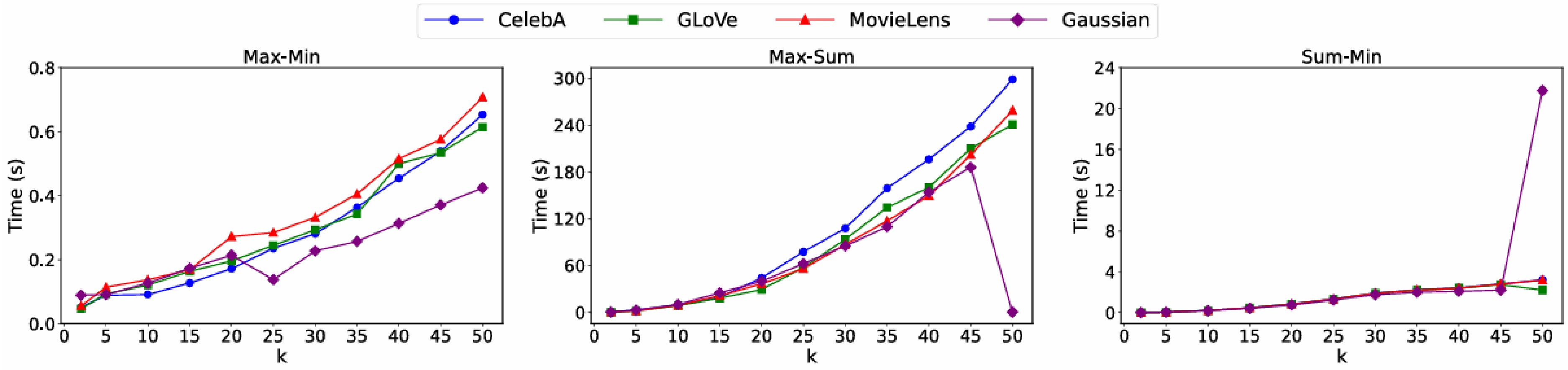


Figure 2: Running time (in seconds) of our algorithms when $k = 2, 5, 10, \ldots, 50$ and $\alpha = 1$.

Thank you for listening!

Ruien Li, Yanhao Wang

School of Data Science and Engineering

East China Normal University