Non-Linear Pricing Restores Tractability for a Data Seller

Eklavya Sharma

PhD Candidate

University of Illinois Urbana-Champaign (UIUC)

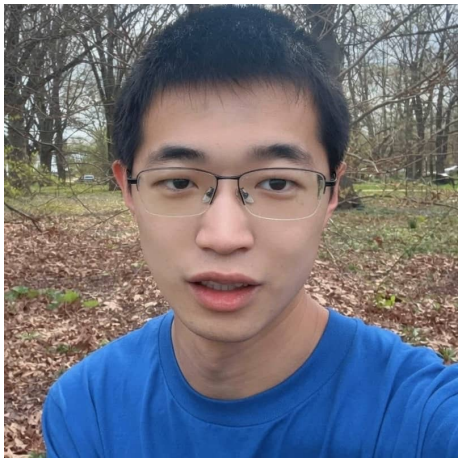
Collaborators



Bhaskar Ray Chaudhury
(UIUC)



Jugal Garg
(UIUC)



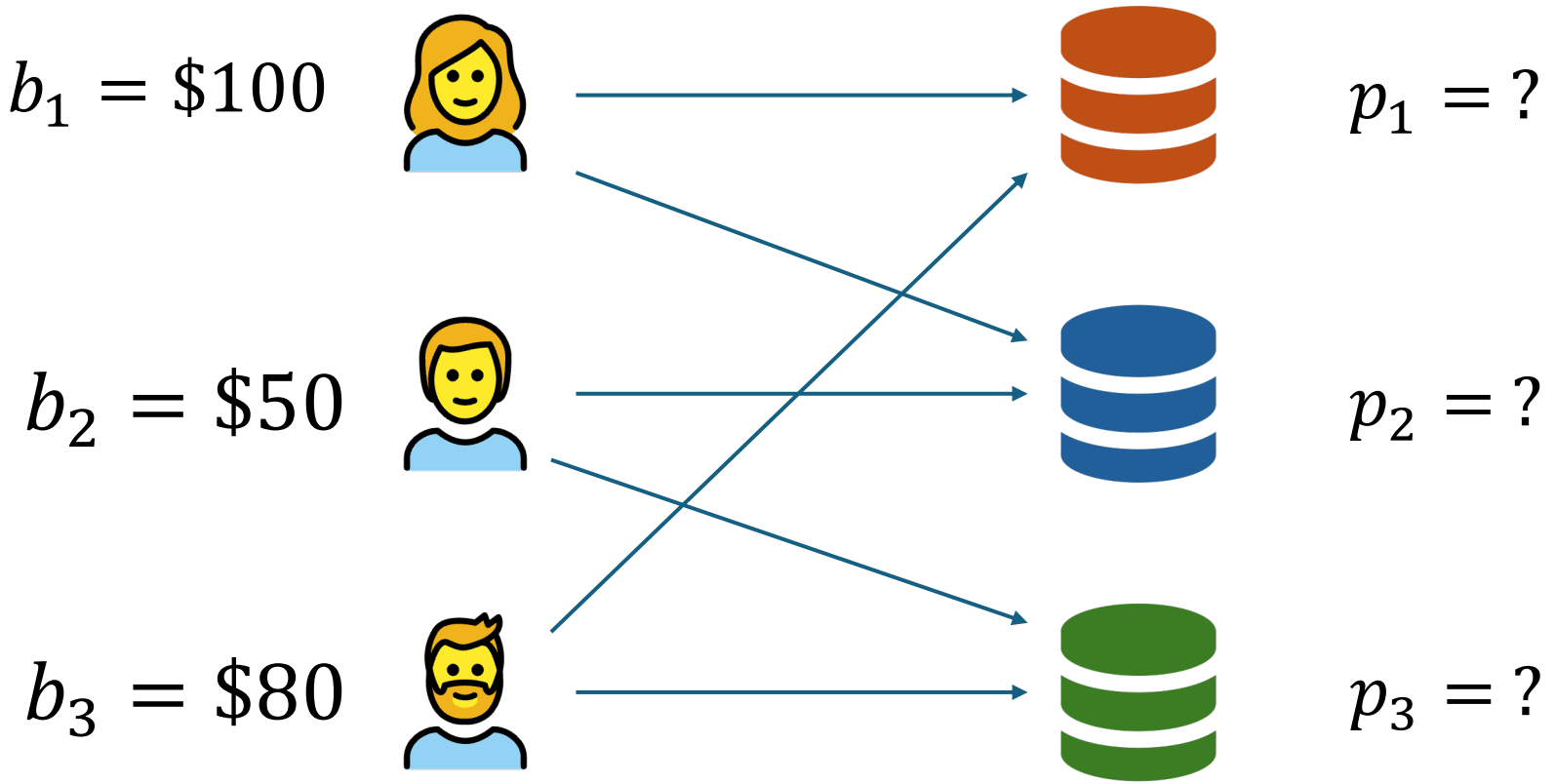
Jiaxin Song
(UIUC)

The Data Economy

- Many firms produce data (passively/actively):
  - e.g., data on real-time foot traffic in retail stores.
- Many firms can use data to improve predictive models:
  - e.g., Uber/Lyft predicting real-time passenger demand.
- Sale of data creates enormous economic value.
- **The problem**: How should a seller price her datasets to maximize revenue?

A model for the sale of data

budgets | n buyers, m datasets | prices



How do buyers purchase?

- Can buy fractions of datasets (subset of rows).
- Bundle of data: represented by $z \in [0,1]^m$, where z_j is fraction of dataset j in the bundle.
- $u_i(z)$ = buyer i's utility for bundle z.
- Under an information-theoretic model of the value of data, $u_i(z)$ is linear in z, so linear utilities are an important special case.
- For prices $p \in \mathbb{R}^m_{\geq 0}$, buyer i selects bundle

$$\underset{z:\, p^T z \leq b_i}{\operatorname{argmax}} \; (u_i(z) - p^T z)$$

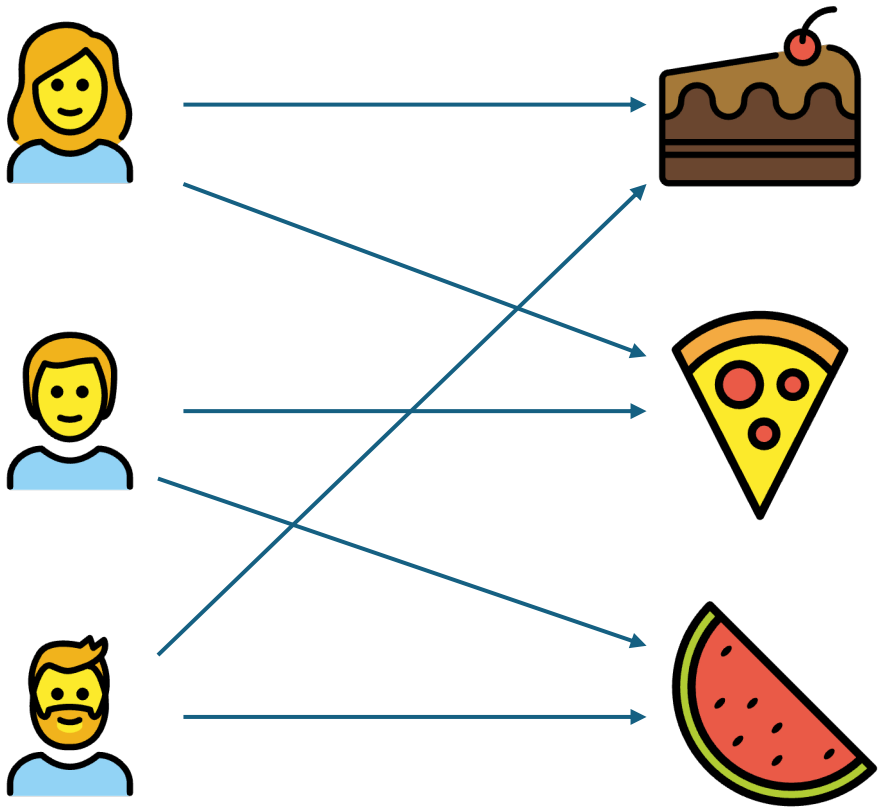
How to maximize revenue?

budgets | n buyers, m goods | prices

$b_1 = \$100$

$b_2 = \$50$

$b_3 = \$80$



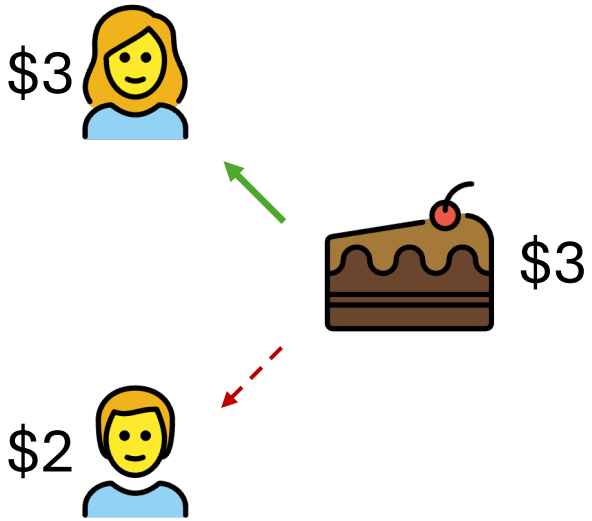
$p_1 = ?$

$p_2 = ?$

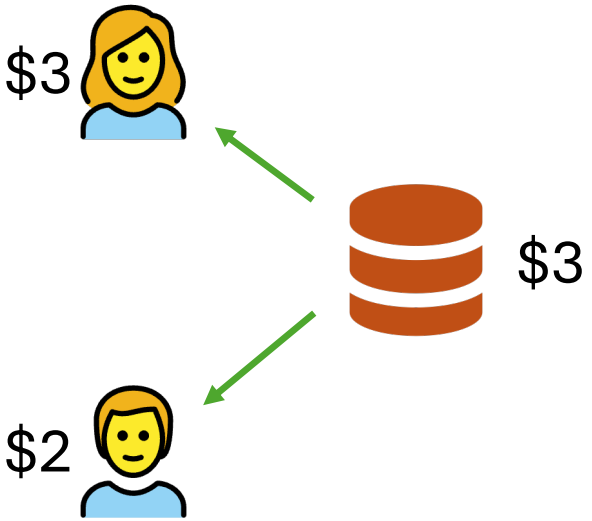
$p_3 = ?$

How these settings differ

Data is *non-rivalrous*: it can be copied and sold to multiple buyers simultaneously.



Revenue = $3



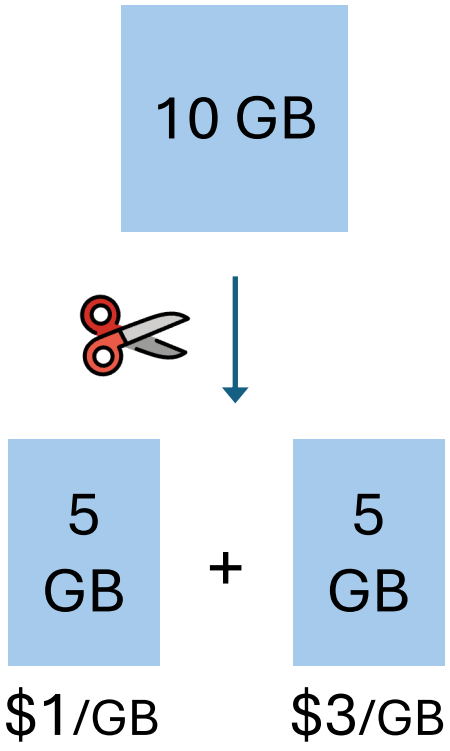
Revenue = $5

Revenue maximization

- Pricing traditional goods has been well-studied. Revenue-maximizing prices can be computed in polynomial time for linear utilities.
- Can we adapt these results to the sale of data?
- Answer: No!
- **Theorem**: When selling data, computing revenue-maximizing prices is APX-hard, even for linear utilities!
- What now?
  - Approximation algorithms: [CGSS'26].

The sharding trick

- Sharding: a practical trick to work around intractability.
- Break dataset into multiple shards. Then price each shard independently.
- New problem: pick shard sizes and prices simultaneously.



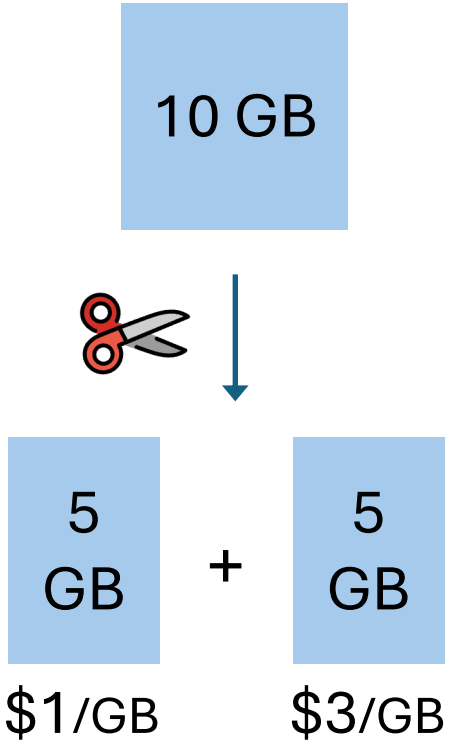
- This new problem is **tractable**; reduces to an LP.
- Sharding can **increase revenue** by up to 2x.



For x fraction of a dataset, pay

$$\begin{cases} x & x \le \frac{1}{2} \\ 3x - \frac{1}{2} & x \ge \frac{1}{2} \end{cases}$$

$\equiv$



Sharding ≡ PLC pricing

PLC: piecewise-linear convex.

Structure of optimal pricing

- What if we use more general pricing functions?
- Answer: *That doesn't help!* PLC is optimal.
- **Theorem**: Given a vector $p = (p_1(\cdot), \ldots, p_m(\cdot))$ of pricing functions, we can construct another vector $\hat{p} = (\hat{p}_1(\cdot), \ldots, \hat{p}_m(\cdot))$ such that:
  - $r(\hat{p}) \geq r(p)$, i.e., $\hat{p}$'s revenue is larger.
  - $\hat{p}_j$ is PLC for all j.

Conclusion

We study the problem “how to set prices for datasets to maximize revenue”. Key takeaways:

1. Non-rivalrous nature of data changes the tractability of revenue maximization, even for linear utilities.
2. Using sharding restores tractability and can raise revenue.
3. Sharding is the optimal way to price.

Thank You



my website



paper

Image credits: openmoji.org (CC BY-SA 4.0)