Analyzing and Optimizing Algorithms for Effective Online and Offline Ad Allocation

Lindsay Kossoff[1,4], Arav Chadha[2,4], Maggie Zhang[2,4], Isabella De La Garza[3,4], Alina Ene[4]

Holton-Arms School, 7303 River Rd, Bethesda, MD 20817[1]; Canyon Crest Academy, 5951 Village Center Loop Rd, San Diego, CA 92130[2]; United High School, 2811 United Ave, Laredo, TX 78045[3]; Boston University, Commonwealth Ave, Boston, MA 02215[4]

Introduction

- With the growth of online advertising, advertisers are seeking more effective strategies to match ads with their target audiences
- This issue can be solved through the generalized assignment problem which aims to find the optimal connections between two sides of a bipartite graph

Bipartite Graph For Ad Allocation

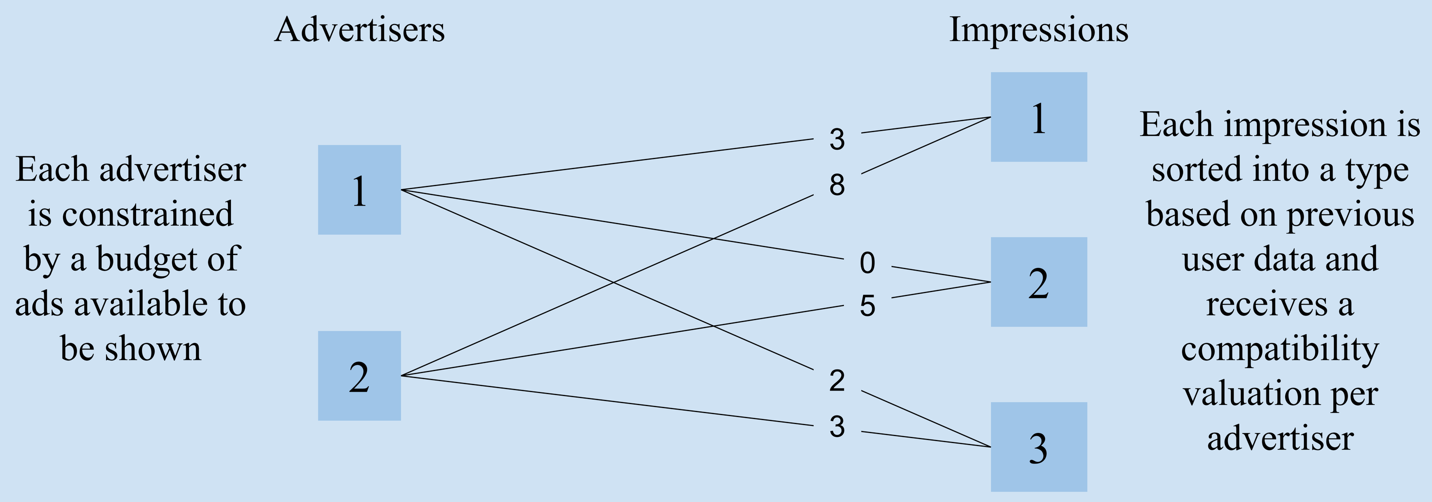

Fig 1. A bipartite graph outlining the nodes, edges and constraints of the ad allocation problem

- There are two types of allocations to consider for our problem
  - Online Ad Allocation: Impressions arrive one at a time with their valuations and need to be allocated immediately.
  - Offline Ad Allocation: All impressions and valuations are known before making allocations.
- While algorithms exist to solve this problem perfectly, they are not viable to run on large data sets due to their extreme run time
- We implemented one algorithms for each type of allocation with the goal of getting considerably close to the optimal solution while keeping a rapid performance.

Methods

Synthetic Instances

- Generates separate lists of advertisers, impressions and weights
  - Every impression, *i*, receives a randomly generated type
  - Each advertiser, *a*, gets a budget and valuation per impression type from an exponential distribution
  - For every advertiser/impression connection, the type-based valuation is added to the weights array

CVXOPT

- Convex optimization algorithm that returns a matrix of optimal allocations

Online Algorithm (Alg. 1):

w_{ai}: valuation of *i* for *a* β_a: threshold value for *a* *a*(EXP): expected advertiser *a*(PRD): predicted advertiser

- Input: Robustness/consistency trade-off parameter $\alpha \in [1, \infty)$
  - For each *a* initialize $\beta_a \leftarrow 0$
  - For each *i*
    - Evaluate expected advertiser using $\max_a \{w_{ai} - \beta_a\}$ and find the predicted advertiser
    - Assign *a* to $\max\{\alpha_B (w_{a(\mathrm{PRD})i} - \beta_{a(\mathrm{PRD})}), w_{a(\mathrm{EXP})i} - \beta_{a(\mathrm{EXP})}\}$
    - Allocate *i* to *a* and if *a* is over budget, remove the least valuable impression of *a*
    - Update β_a for *a*
- Output: Matrix of binary allocations

Offline Algorithm (Alg. 2):

- Inputs: Values λ and ϵ, number of rounds *R*
  - Assign priority score $\beta_a = (1 + \epsilon)^{-R}$ for all *a*
  - For rounds *R*
    - For all *i* in *a*: Update fractional allocation of *i* to *a*
    - For all *a*: If *a* is under or over budget then update β_a
  - For all *a* over budget, remove least valuable *i* until budget is met
- Output: Matrix of fractional allocations

If $\sum_{a'} D_{a,i,\lambda} \leq 1$, allocate $D_{a,i,\lambda}$
Else allocate $D_{a,i,\lambda} / \sum_{a'} D_{a,i,\lambda}$

$D_{a,i,\lambda} = \beta_a \cdot e^{(w_{a,i} - 1)/\lambda}$
a' = all *a*

If $\mathrm{Alloc}_a \leq b_a / (1+\epsilon)$ then $\beta_a := (1+\epsilon) \cdot \beta_a$
If $\mathrm{Alloc}_a \geq (1+\epsilon) \cdot b_a$ then $\beta_a := \beta_a / (1+\epsilon)$

Alloc_a = total *i* allocated to *a*
b_a = budget of *a*

Testing:

- Fine-Tuning: We used a heatmap to find the optimal values of λ and ϵ for Alg. 2
- Threshold Implementation: We compared the objective value found and run time for an exponential average, uniform average, and lowest weight threshold for Alg. 1
- Overall Comparison: For each data type we compared the run time (in seconds) and objective value found on instances of various sizes
- Data Types: Each algorithm ran on synthetic instances, corrupted instances, and real world data from the Stanford Large Network Dataset Collection

Results

Fine-Tuning and Threshold Implementation

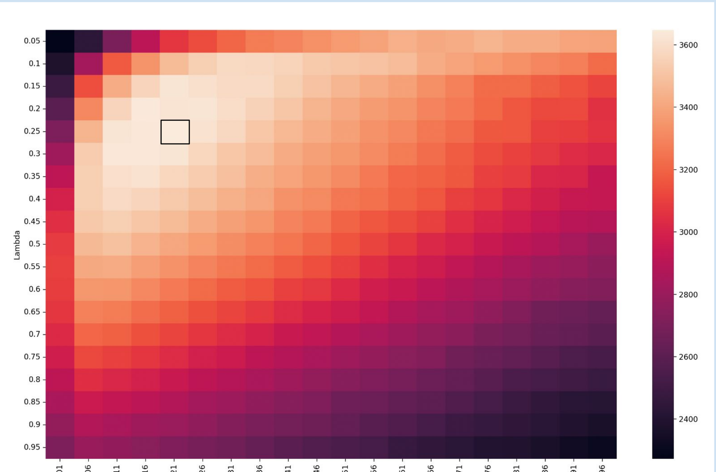

Fig 2. A heatmap displaying the objective value achieved through different combinations of epsilon and lambda (50 advertisers, 1000 impressions, and 50 rounds)

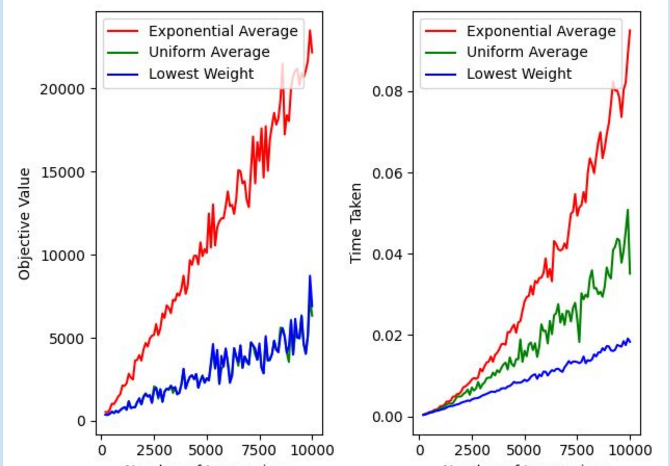

Fig 3. Graphs comparing the objective value and time taken for three separate Alg. 1 threshold calculations (100 advertisers with increments of 100)

Comparison on Synthetic Data

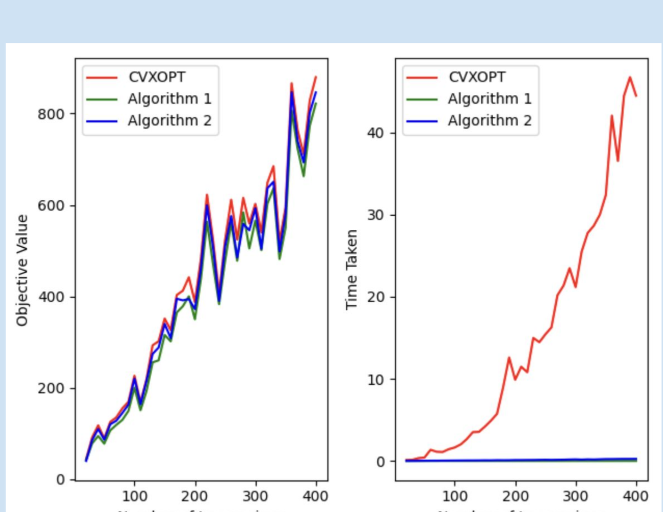

Fig 4. Objective value and run time of Alg. 1, Alg. 2, and CVXOPT (20 advertisers with increments of 10)

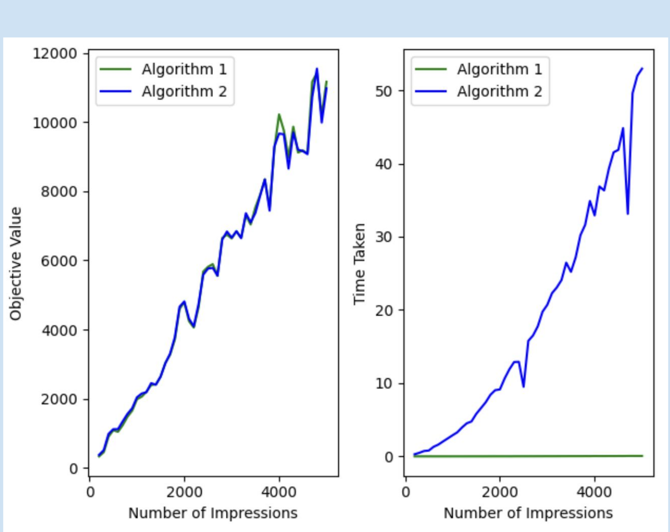

Fig 5. Objective value and run time of Alg. 1 and Alg. 2 (50 advertisers with increments of 100)

Comparison on Other Data Types

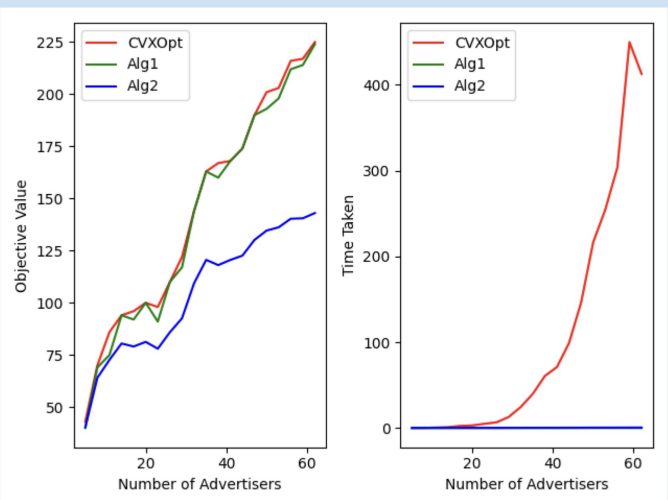

Fig 6. Objective value and run time of Alg. 1, Alg. 2, and CVPXOPT on real-world data from the Stanford Large Network Dataset Collection (impressions determined by number of advertisers with increments of 3)

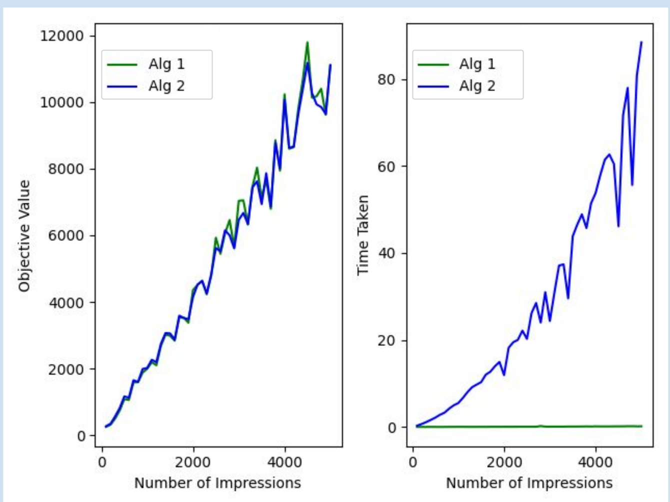

Fig 7. Objective value and run time of Alg. 1 and Alg. 2 on corrupted synthetic data (20 advertisers with increments of 100)

Conclusion

Fine-Tuning and Threshold Implementation:

- Using Fig. 2, we identified λ = 0.25 and ϵ = 0.21 as the optimal parameters for Alg 2
- Fig. 3 shows that the exponential average calculates the best allocations for Alg 1 with a negligible time difference compared to the other threshold update methods

Comparisons on Different Data Types:

- Alg. 1 and Alg. 2 consistently found nearly optimal solutions in a viable time for large data sets on synthetic and corrupted synthetic data.
- Fig. 6 shows Alg. 1 outperforming Alg. 2 on binary real-world data, highlighting the strength of Alg. 1's binary allocations

Future Work:

- Implement machine learning predictions rather than mathematical for Alg. 1
- Utilize algorithms for other bipartite graph problems with similar constraints

Acknowledgements

I would like to express my deepest gratitude to Dr. Ene for her invaluable and supportive mentorship throughout this research. I would also like to give a special thank you to my labmates. I could not imagine doing this project without your collaboration and encouragement. Lastly, I would like to thank my parents and RISE for providing me with this remarkable opportunity.

References

Agrawal, S.; Zadimoghaddam, M.; Mirrokni, V. Proportional Allocation: Simple, Distributed, and Diverse Matching with High Entropy. *PMLR* [Online], 2018, 80: 99-108 https://proceedings.mlr.press/v80/agrawal18b.html (accessed 2024-08-06).

Andersen, M.; Dahl, J.; Vandenberghe, L. CVXOPT Version 1.3 2023

Leskovec, J.; Krevl, A. Wikipedia Adminship Election Data, 2008. Stanford Large Network Dataset Collection. https://snap.stanford.edu/data/ (accessed 2024-08-06).

Spaeh, F.; Ene, A. Online Ad Allocation with Predictions. *arXiv* [Online], May 24, 2023, https://arxiv.org/pdf/2302.01827 (accessed 2024-08-06).