Pareto-Weighted-Sum-Tuning: Learning-to-Rank for Pareto Optimization Problems

Harry Wang, - Computer Science and Engineering
Brian Denton, PhD - Industrial and Operations Engineering

University of Michigan-Ann Arbor

Pareto Optimization

- Multiobjective and Multicriteria Optimization
- Criteria often conflict with one another
- Need to consider trade-offs
- Common in decision-making problems

Pareto Optimization

Choice A
- Very High Best Case Performance
- Very Low Worst Case Performance

Choice B
- Lower Best Case Performance
- Higher Worst Case Performance

- Different Objective Functions
- Trade-offs based on objective function used
- How does DM evaluate trade-offs?

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Applications/Domains

- Mechanical/Civil/Structural Engineering
  - Cost vs. Strength
- Medicine/Healthcare
  - Efficacy vs. Side-effects
- Finance/Business
  - Risk vs. Reward

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Mathematical Formulation

- Weighted Sum Approach:

$$\max \sum_{i=1}^{m} \alpha_i c_i$$

$$s.t. \sum_{i=1}^{m} \alpha_i \leq 1$$

α_i = weight on criteria i
c_i = quantity of criteria i

- Common Problem: Figuring out these weights is difficult
- Can rely heavily on heuristics or domain knowledge

(Yang)

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Previous Work

- Analytical Hierarchy Process (AHP)
  - Compare and Rate Criteria Nodes
  - Requires Explicit Quantifying from Decision-Maker

  (Dyer)

- Multinomial-Logit Model
  - Analyze Users' Choices to determine preferences
  - Successful in Marketing, Natural Language Processing, Computer Vision

  (Negahban, Oh, Thekumparampil, and Xu)

Methodology Developed

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Outline of Approach

- Give decision maker specific instances of Objective Value tuples $[t_1, t_2, \ldots, t_x]$

  $t_i = [c_{1i}, c_{2i}]$

- Decision maker ranks tuples → simpler than scoring
- Use Supervised Learning to learn weights based on decision maker's feedback

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Pareto-Weighted-Sum-Tuning (PWST)

Algorithm 1 Pareto-Weighted-Sum-Tuning

Require: n data-points, x data-points at each iteration where $x \leq n$

α-vectors $= \emptyset$

$i = 0$

$[t_1, t_2, ..., t_x] \leftarrow x$ data-points sampled from the data-set and displayed to decision-maker in random order

while $i \leq \lfloor \frac{n}{x} \rfloor$ **do**

(I) $[r_1, r_2, ..., r_x] \leftarrow$ ranking given by decision-maker for the x sampled data-points; this action is equivalent to "correcting" the order predicted by the algorithm

(II) $[\alpha_1, \alpha_2, ..., \alpha_m] \leftarrow$ Ranking-SVM($[t_1, t_2, ..., t_x], [r_1, r_2, ..., r_x]$)

(III) $[t_1, t_2, ..., t_x] \leftarrow x$ data-points sampled from the data-set (Note: in this step, Pareto-Weighted-Sum-Tuning uses the α-values learned to predict r_{true} and displays $[t_1, t_2, ..., t_x]$ according to its predicted order based on $\sum_{i=1}^{m} \alpha_i c_i$, where a higher sum corresponds to a better rank)

end while

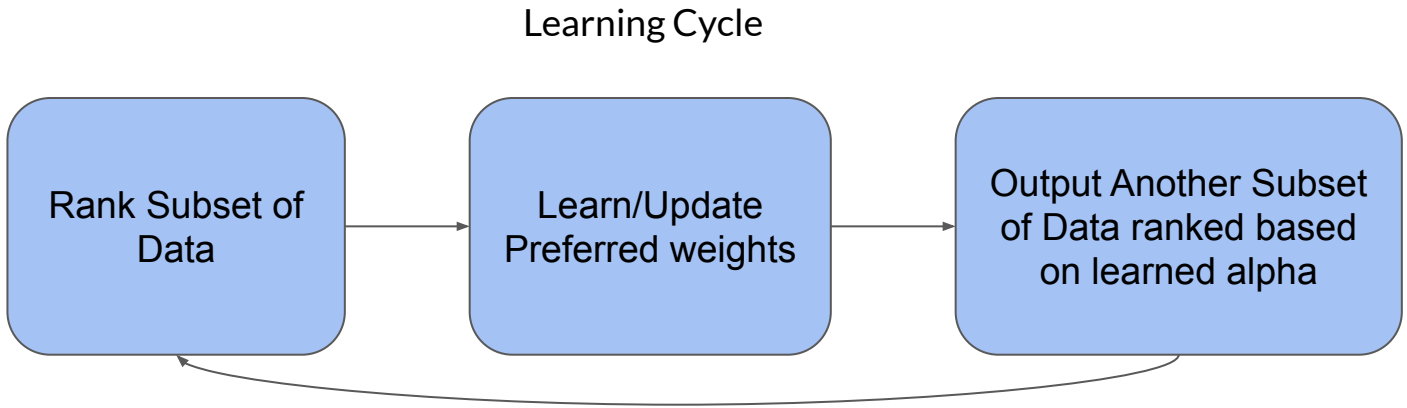
$\vec{\alpha} = [\bar{\alpha_1}, \bar{\alpha_2}, ..., \bar{\alpha_m}]$ where $\bar{\alpha_i}$ is the mean value of α_i of vectors in α-vectors

return $\vec{\alpha}$

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

PWST

- Give DM a subset of data
- Example: given n tuples, label from 0-n
- 0 → worst pair; n→ best pair
- Learn latent preferred weights with Ranking-SVM



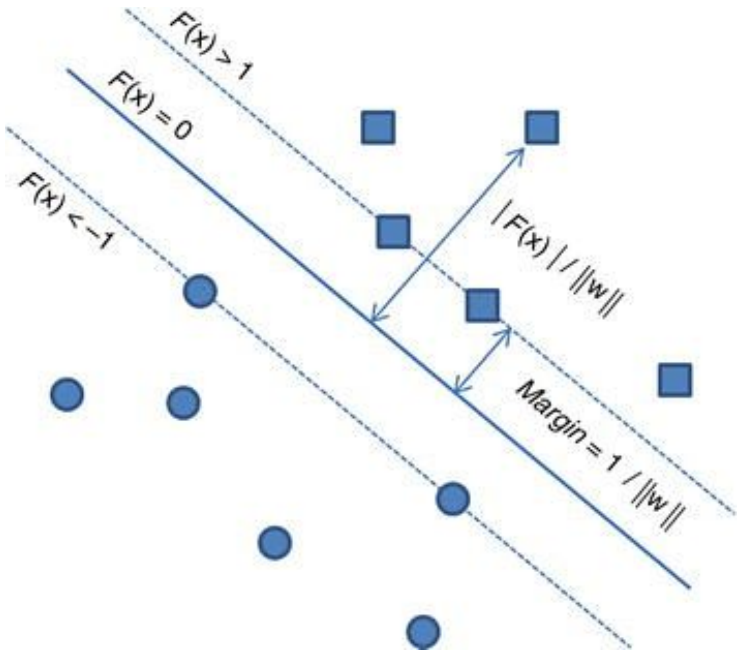
MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Learn-to-Rank

- Ranking-Support-Vector-Machine
- Learn ranking function f that fits rankings
- Optimize following function:

$$\min -\tau(r_{true}, r_{pred})$$

(Joachims)



MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Kendall Tau

- Compare true ranking with predicted ranking
- Kendall Tau:

$$\tau(r_{true}, r_{pred}) = \frac{A - B}{A + B}$$

- A - Number of ordered pairs in both r_{true}, r_{pred}
- B - Number of ordered pairs in both r_{pred}, but not r_{true}

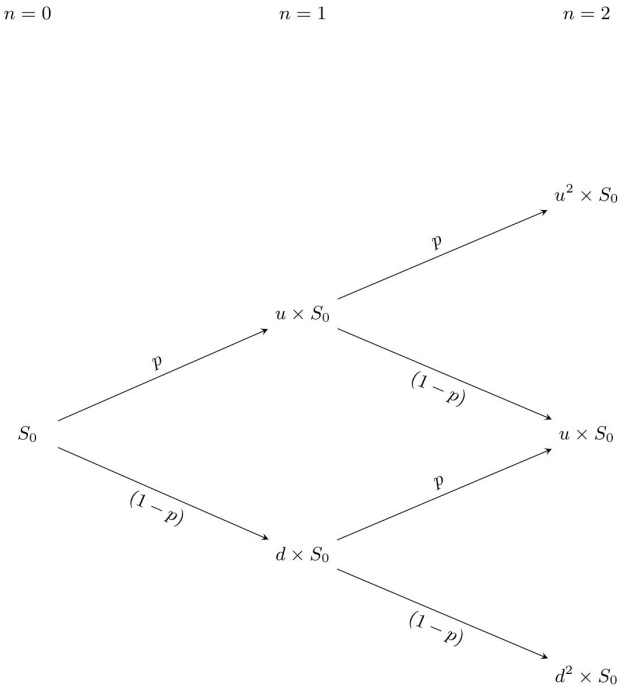
(Joachims)

Example Application

PWST Application Example

- Stock Investment
- Volatile and Hard to Predict
- Binomial Pricing Tree Model

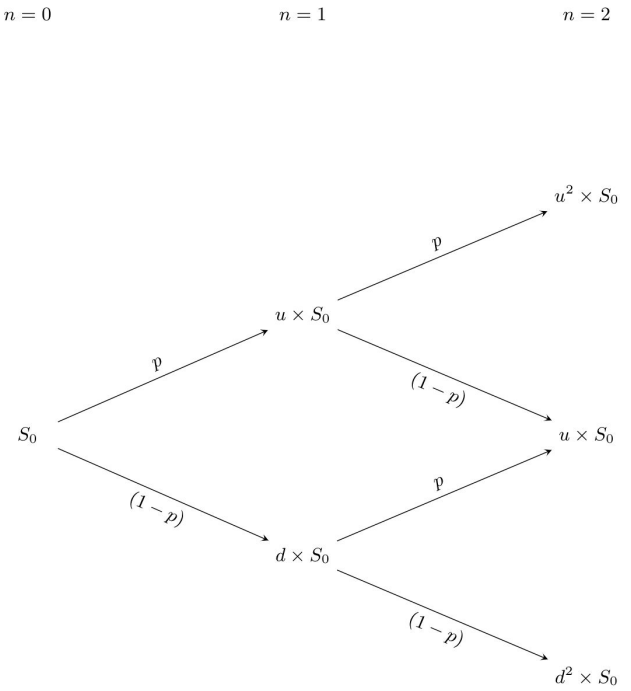
(Muzzioli, S. and Torricelli)

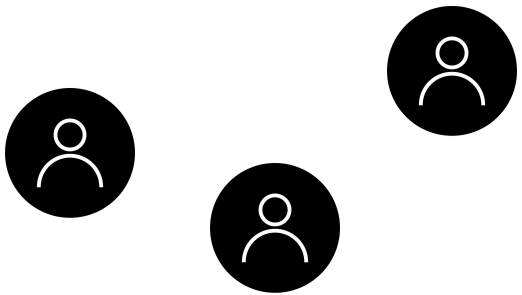


MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

PWST Application Example

- Financial Advisor
- Different Financial Clients
- Different preferences
- Hard to express explicitly

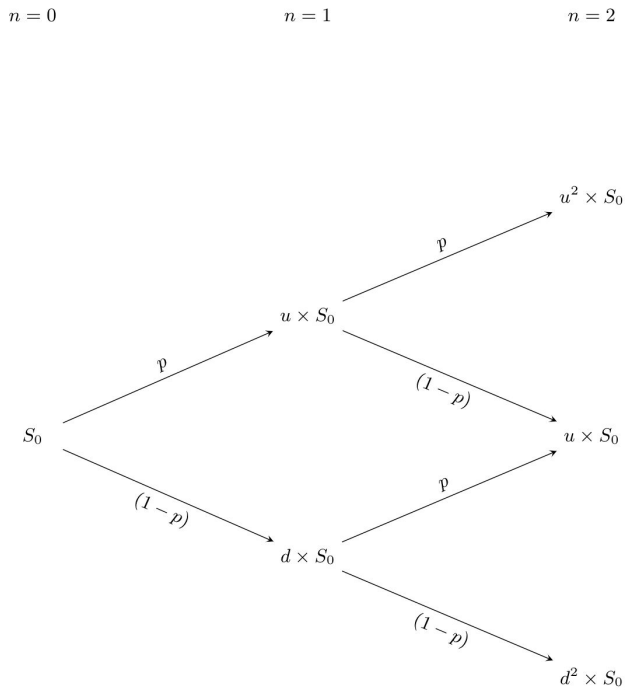




MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Application Example

- o_i = optimistic expected value
- p_i = pessimistic expected value
- α = client's weight on o
- s_i = price of stock i
- b = budget limit
- x_i = amount of stock i purchased



Application Example

- Expected utility of purchasing stock i:

$$f_i(\alpha) = \alpha \times o_i + (1 - \alpha) \times p_i$$

- Optimization Problem:

$$\max \sum_{i=1}^{s} x_i f_i(\alpha)$$

$$s.t. \sum_{i=1}^{s} s_i x_i \leq b$$

$$x_i \in \{0, 1\}$$

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Application Example

- Must know α to solve problem
- We have data-set of previous stock prices
- Use PWST to find α
- Use integer programming methods to solve once α is known

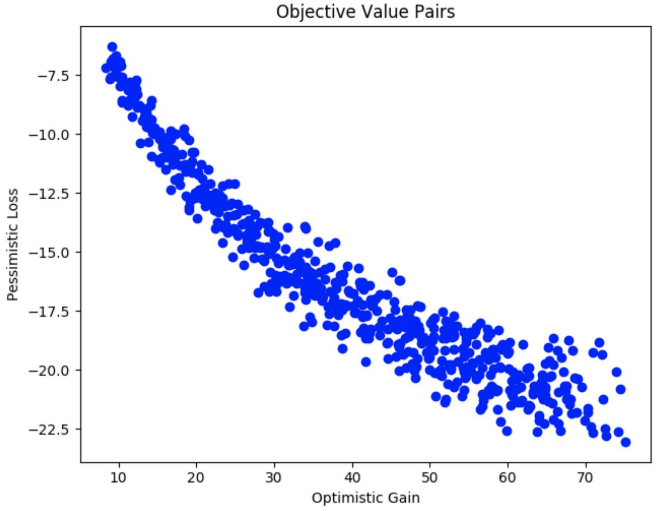
$$\max \sum_{i=1}^{s} x_i f_i(\alpha)$$

$$s.t. \sum_{i=1}^{s} s_i x_i \leq b$$

$$x_i \in \{0, 1\}$$

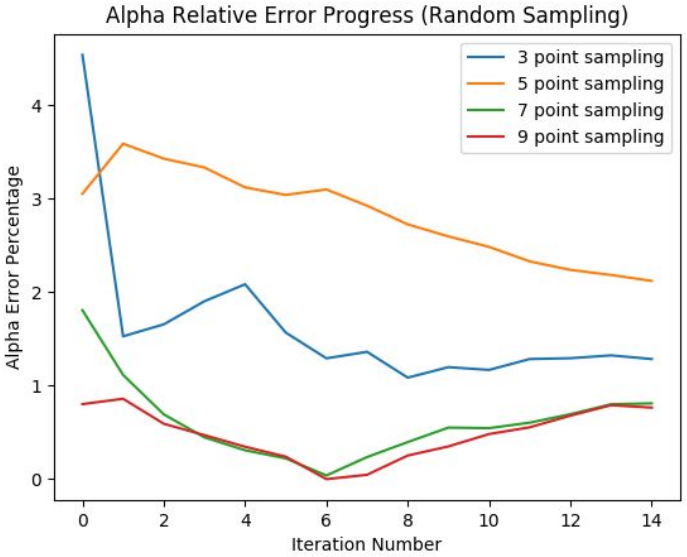
Data Collection: Simulated Decision-Maker

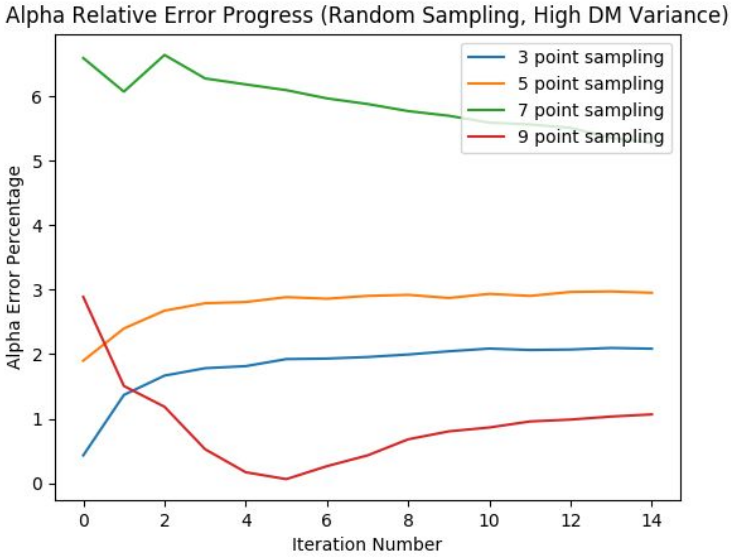
- Used Binomial Pricing Tree Model to generate data (with noise)
- Python Program to simulate DM
- Initialized with α value
- Given variance level δ
- DM draws from $\text{unif}(\alpha, \delta)$ on each iteration



MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Results with Virtual DM





MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Results with Virtual DM

- Some configurations achieve a relative error of below 1 percent after 15 iterations
- Viable approach to further explore
- Can help with weight-tuning problem in Pareto Optimization

Thank you!

Sources

Ahmadian, S., Bhaskar, U., Sanità, L. and Swamy, C., 2018. Algorithms for Inverse Optimization Problems. In: 26th Annual European Symposium on Algorithms. Dagstuhl Publishing, Germany.

Beck, J., Chan, E., Irfanoglu, A. and Papadimitriou, C., 1999. Multi-criteria optimal structural design under uncertainty. Earthquake Engineering and Structural Dynamics, 28(7), pp.741-761.

Bertsimas, D. and Demir, R., 2002. An Approximate Dynamic Programming Approach to Multidimensional Knapsack Problems. Management Science, 48(4), pp.550-565.

Bettinelli, A., Cacchiani, V. and Malaguti, E., 2017. A Branch-and-Bound Algorithm for the Knapsack Problem with Conflict Graph. INFORMS Journal on Computing, 29(3), pp.457-473.

Bruch, S., 2020. An Alternative Cross Entropy Loss For Learning-To-Rank. [online] arXiv.org. Available at: https://arxiv.org/abs/1911.09798 [Accessed 27 May 2020].

Sources

Bärmann, A., Martin, A., Pokutta, S. and Schneider, O., 2020. An Online-Learning Approach To Inverse Optimization. [online] arXiv.org. Available at: \url{https://arxiv.org/abs/1810.12997} [Accessed 27 May 2020].

Chang, K., 2016. E-Design. Amsterdam: Elsevier.

Dong, C., Chen, Y. and Zeng, B., 2018. Generalized Inverse Optimization through Online Learning. In: 32nd Conference on Neural Information Processing Systems (NeurIPS 2018). Montréal, Canada.

Dyer, J., 1990. Remarks on the Analytic Hierarchy Process. Management Science, 36(3), pp.249-258.

Engau, A. and Sigler, D., 2020. Pareto solutions in multicriteria optimization under uncertainty. European Journal of Operational Research, 281(2), pp.357-368.

Gensch, D. and Recker, W., 1979. The Multinomial, Multiattribute Logit Choice Model. Journal of Marketing Research, 16(1), p.124.

Sources

Ghobadi, K., Lee, T., Mahmoudzadeh, H. and Terekhov, D., 2018. Robust inverse optimization. Operations Research Letters, 46(3), pp.339-344.

Heuberger, C., 2004. Inverse Combinatorial Optimization: A Survey on Problems, Methods, and Results. Journal of Combinatorial Optimization, 8(3), pp.329-361.

Joachims, T., 2002. Optimizing Search Engines using Clickthrough Data. In: KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp.133-142.

Liang, X., Zhu, L. and Huang, D., 2017. Multi-task ranking SVM for image cosegmentation. Neurocomputing, 247, pp.126-136.

Marler, R. and Arora, J., 2009. The weighted sum method for multi-objective optimization: new insights. Structural and Multidisciplinary Optimization, 41(6), pp.853-862.

Sources

Muzzioli, S. and Torricelli, C., 2005. The pricing of options on an interval binomial tree. An application to the DAX-index option market. European Journal of Operational Research, 163(1), pp.192-200.

Negahban, S., Oh, S., Thekumparampil, K. and Xu, J., 2018. Learning from Comparisons and Choices. Journal of Machine Learning Research, 19(1-95).

Shvimer, Y. and Herbon, A., 2020. Comparative empirical study of binomial call-option pricing methods using S and P 500 index data. The North American Journal of Economics and Finance, 51, p.101071.

Wang, H., 2020. Pareto-Weighted-Sum-Tuning.\url{https://github.com/harryw1248/Pareto-Weighted-Sum-Tuning}.

Yang, X., 2016. Nature-Inspired Optimization Algorithms. [Place of publication not identified]: Elsevier.