A Multi-objective / Multi-task Learning Framework Induced by Pareto Stationarity

Michinari Momma[1], Chaosheng Dong[1], Jia Liu[2]

[1]Amazon.com, [2]The Ohio State University

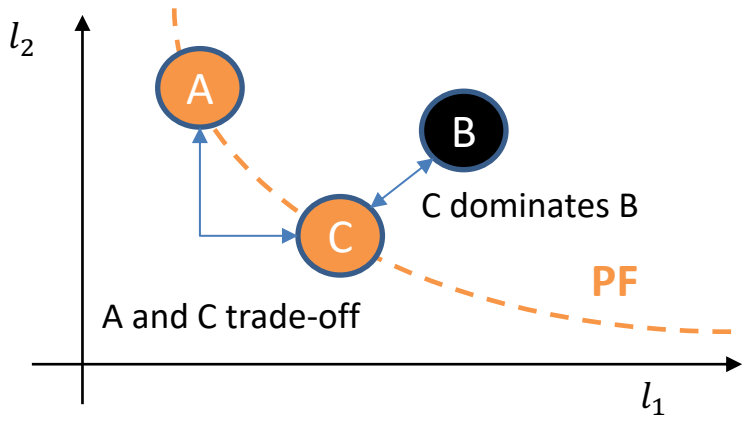
amazon

Introduction

- MOO / MTL has become very popular modeling technique
  - MOO: Product search models are trained with multi-objectives
  - MTL: Popular way to training deep learning models
- Trade-off: common property in MOO / MTL
  - Many models with different trade-offs
  - Need to specify preference to select a model for experiments, deployment, etc.
- Challenges
  1. How to explore (all) trade-offs over multi-objectives / tasks
  2. How to train a (specific) model that aligns with user preference
  3. How to improve all objective / tasks when re-train / fine-tune existing models

MOO / MTL problem setting

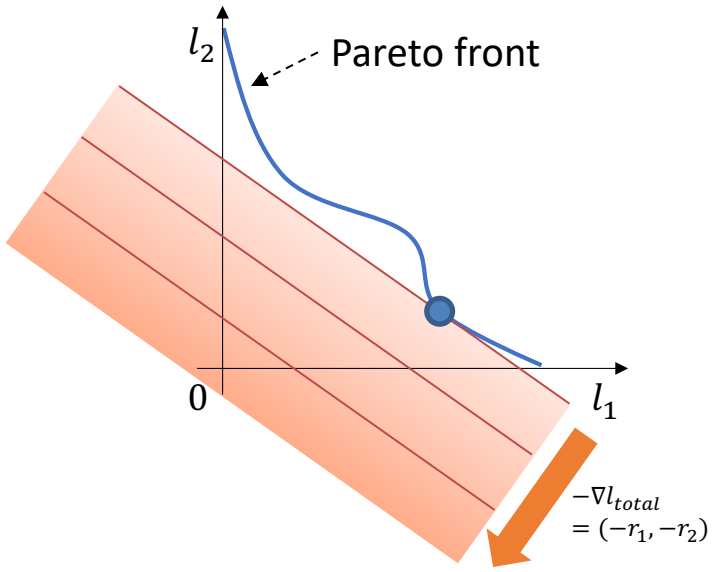
- Given a model parameter vector $\boldsymbol{x} \in R^n$, loss for m objectives / tasks: $l_i(\boldsymbol{x}), i = 1, \ldots, m$
- Goal: min $\boldsymbol{l}(\boldsymbol{x}) = [l_1(\boldsymbol{x}), \ldots, l_m(\boldsymbol{x})]$ $(n \gg m)$
- Solutions typically have **trade-offs**, where there is no better solution (i.e., **dominates**) – **Pareto optimal** (PO) solution
- **Pareto Front** (PF) is a set of PO solutions.

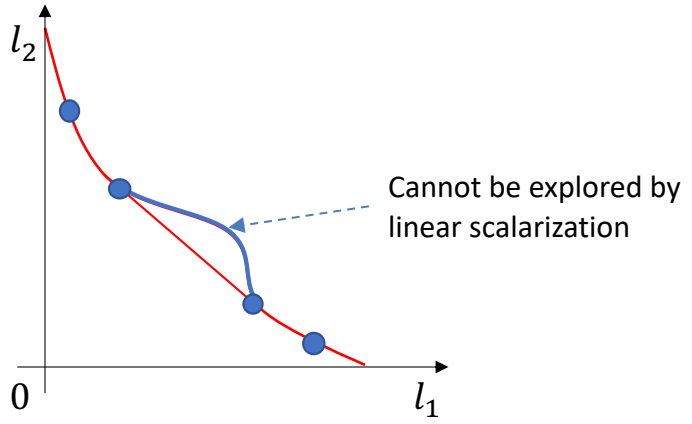


amazon

Example of preferences (Linear scalarization)

- Linear scalarization ($l_{LS} = r_1 l_1 + r_2 l_2$)
  - Higher r, the smaller l.
  - "level set" $r_1 l_1 + r_2 l_2 = const$, on a tangential hyperplane
  - Cannot explore non-convex portion of PF





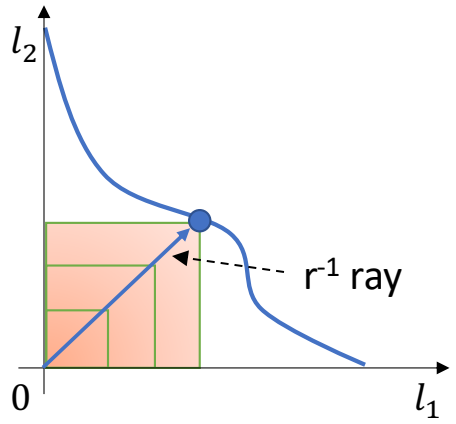
amazon

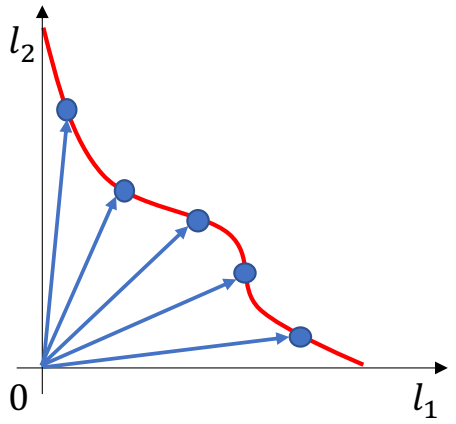
Example of preference (weighted Chebyshev)

- Ratio between loss ($r_1 l_1 = r_2 l_2$) – "weighted Chebyshev (WC)"
  - Higher r, the smaller l.
  - Level set on a hyper-rectangular box:

$$r_1 l_1 = r_2 l_2 = c \Rightarrow (l_1, l_2) = \left(\frac{c}{r_1}, \frac{c}{r_2}\right) : \boldsymbol{r}^{-1}\text{ray}$$

  - Can explore full PF by various $\mathbf{r}^{-1}$ rays
  - EPO Search (ICML 2020) solves WC

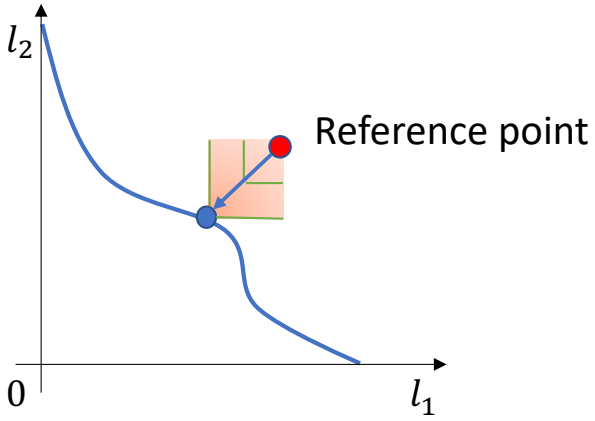


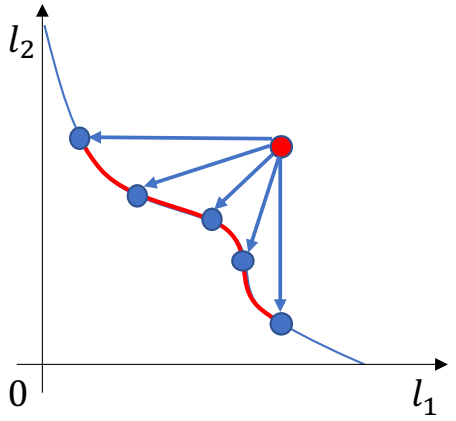


amazon

Example of preference (Reference point)

- Preference from Reference point – Extended Weighted Chebyshev (XWC)
  - Designed to find solution better than the reference point
  - Can explore a specific portion of PF, pivoted on the reference point
  - Useful for model retraining / finetuning on an existing baseline / pretrained model





amazon

Gradient based method to solve PO: MGDA

- Multiple Gradient Descent Algorithm (MGDA):
  - Gradient matrix: $\boldsymbol{K} = f(\nabla l_1, \dots, \nabla l_m) \in R^{m \times m}$
  - Pareto stationarity:

$$\boldsymbol{K\alpha} = 0, \boldsymbol{e^T\alpha} = 1, \boldsymbol{\alpha} \geq \boldsymbol{0}, \boldsymbol{e} = [1, \dots, 1]$$

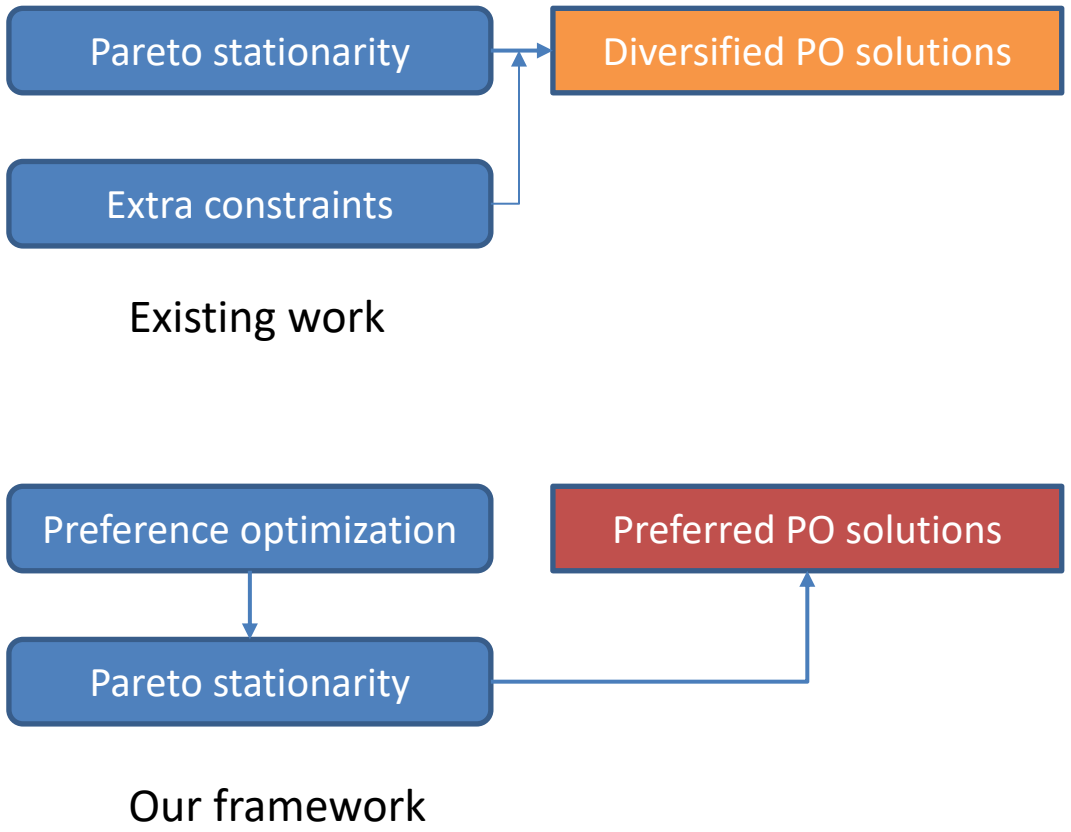
  - MGDA solves: $\min_{\boldsymbol{\alpha}} \|\boldsymbol{K\alpha}\|_2, \quad s.t. \boldsymbol{e^T\alpha} = 1, \boldsymbol{\alpha} \geq \boldsymbol{0}$
  - Once we find $\boldsymbol{\alpha}$, run gradient descent

$$x^{new} := x^{old} - \eta \boldsymbol{G\alpha}$$

  - [challenge] Although MGDA finds any PO, **no control** on what solution to discover.

Improvement over MGDA

- Existing approaches try to modify MGDA (i.e. Pareto stationarity) by adding extra constraints
  - PMTL (Lin et al., NeurIPS 2019), etc.
- Note Pareto stationarity itself is a part of optimality condition in “some” optimization problem
- We formulate “preference” first, then derive Pareto stationarity



amazon

Implementation of WC: WC-MGDA

- Weighted Chebyshev problem: ℓ_∞-norm minimization of weighted loss functions

$$\min \rho \quad s.t. \quad \boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x}) \leq \rho \boldsymbol{e}$$

- By KKT conditions and $\boldsymbol{K_r} = \mathrm{diag}(\boldsymbol{r})\boldsymbol{K}\mathrm{diag}(\boldsymbol{r})$, Wolfe Dual is

$$\begin{aligned} &\max \boldsymbol{\alpha}^T(\boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x})) \\ &s.t. \quad \boldsymbol{e}^T\boldsymbol{\alpha} = 1, \boldsymbol{\alpha} \geq 0, \boldsymbol{K_r\alpha} = \boldsymbol{0} \end{aligned}$$

- Like MGDA, we minimize ℓ_2-norm of $\boldsymbol{K_r\alpha}$:

$$\begin{aligned} &\max \boldsymbol{\alpha}^T\big(\boldsymbol{r} \odot \boldsymbol{l}(\boldsymbol{x})\big) - u\gamma \\ &s.t. \quad \boldsymbol{e}^T\boldsymbol{\alpha} = 1, \boldsymbol{\alpha} \geq 0, \|\boldsymbol{K_r\alpha}\|_2 \leq \gamma \end{aligned}$$

"MGDA" portion to optimize Pareto stationarity

  - "trade-off" u is auto-tuned

amazon

Formulation of XWC: XWC-MGDA

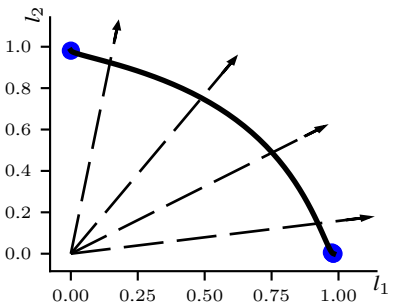
- Extend WC by two modifications:
  - Including reference point ($\boldsymbol{b}$) to the formulation
  - Setting lower bound of $\boldsymbol{\alpha}$ to ensure strict optimality ($\boldsymbol{\alpha} \geq \boldsymbol{w}$)
- The XWC-MGDA problem is;

$$\max \boldsymbol{\alpha}^T(\boldsymbol{r} \odot (\boldsymbol{l}(\boldsymbol{x}) - \boldsymbol{b})) - u\gamma$$
$$s.t. \quad \boldsymbol{e}^T\boldsymbol{\alpha} = 1, \boldsymbol{\alpha} \geq \boldsymbol{w}, \|\boldsymbol{K_r}\boldsymbol{\alpha}\|_2 \leq \gamma$$

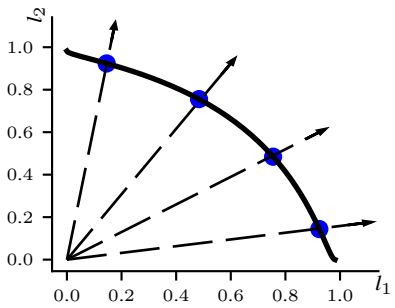
Experimental results

Experiment on non-convex synthetic data:

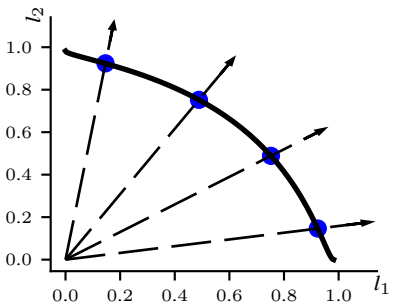
$$\boldsymbol{l}(\boldsymbol{x}) = \left[1 - \exp\left(-\left\|\boldsymbol{x} - \frac{1}{\sqrt{n}}\boldsymbol{e}\right\|^2\right), 1 - \exp\left(-\left\|\boldsymbol{x} + \frac{1}{\sqrt{n}}\boldsymbol{e}\right\|^2\right)\right]$$



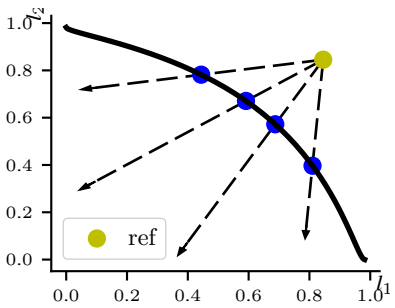
LinScalar



EPO



WC-MGDA



XWC-MGDA
(w/ ref point)

amazon

Summary

- Proposed a new framework to formulate preferences in MOO / MTL
- XWC-MGDA allows us to explore PF pivoted on a given reference point
- Experimental results (see main paper) quantify competitive performance of XWC-MGDA

amazon