Learning the Pareto Front with Hypernetworks

Polina Barabanshchikova

MIPT

March 14, 2023

Multi-objective Optimization (MOO)

- MOO is defined by m losses $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_m)$, $\ell_i : \mathbb{R}^d \to \mathbb{R}_+$
- A partial ordering is defined on the loss space

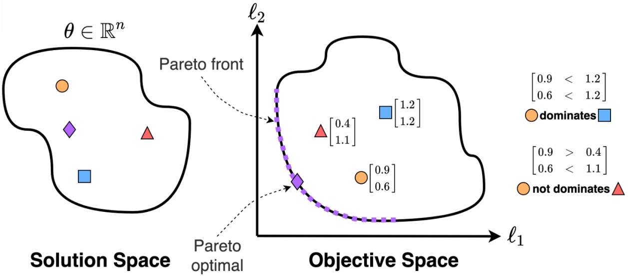
$$\boldsymbol{\ell}(\theta_1) \preceq \boldsymbol{\ell}(\theta_2) \text{ if } \ell_i(\theta_1) \leq \ell_i(\theta_2) \text{ for all } i$$

- A point $\theta_1 \in \mathbb{R}^d$ dominates $\theta_2 \in \mathbb{R}^d$ if $\boldsymbol{\ell}(\theta_1) \prec \boldsymbol{\ell}(\theta_2)$, that is

$$\boldsymbol{\ell}(\theta_1) \preceq \boldsymbol{\ell}(\theta_2) \text{ and } \ell_i(\theta_1) < \ell_i(\theta_2) \text{ for some } i$$

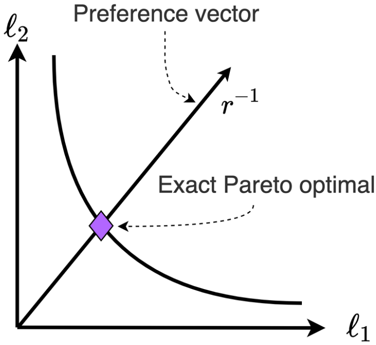
Pareto Optimality

- *Pareto optimal point* – not dominated by any other point
- *Pareto front* – set of all Pareto optimal solutions



Exact Pareto Optimality

- Each optimal point is an intersection of the front and desired direction in loss space – a **preference vector**
- Preference vector represents a trade-off between objectives
- Given a preference vector the goal is to find an optimal solution on that preference



Limitations of Previous Methods

Previous MMO approaches have the following drawbacks

- **Scalability**: A separate model has to be trained for each point on the front. The number of models to be trained to cover the objective space grows exponentially with the number of objectives.
- **Flexibility**: The decision maker cannot switch freely between preferences unless all models are trained and stored in advance.

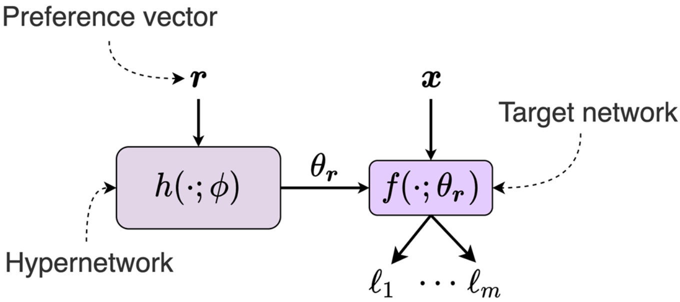
Pareto Front Learning

The goal is to design a **single** model that can be applied at inference time to **any given preference** direction, even ones not seen during training.

- Scalability: Train and store a single model
- Flexibility: Switch trade-off points during inference

Pareto HyperNetworks (PHN)

Pareto HyperNetwork $h(\cdot;\psi)$ receives an input preference ray $\mathbf{r}$ and outputs the corresponding Pareto optimal model weights $\theta_{\mathbf{r}}$.



Pseudocode

Algorithm 1 PHN

---

while not converged **do**
  $\boldsymbol{r} \sim Dir(\boldsymbol{\alpha})$
  $\theta(\phi, \boldsymbol{r}) = h(\boldsymbol{r}; \phi)$
  Sample mini-batch $(x_1, y_1), .., (x_B, y_B)$
  if LS **then**
    $g_\phi \leftarrow \frac{1}{B} \sum_{i,j} r_i \nabla_\phi \ell_i(x_j, y_j, \theta(\phi, \boldsymbol{r}))$
  if EPO **then**
    $\beta = EPO(\theta(\phi, \boldsymbol{r}), \boldsymbol{\ell}, \boldsymbol{r})$
    $g_\phi \leftarrow \frac{1}{B} \sum_{i,j} \beta_i \nabla_\phi \ell_i(x_j, y_j, \theta(\phi, \boldsymbol{r}))$
  $\phi \leftarrow \phi - \eta g_\phi$
return ϕ

- PHN-LS uses linear scalarization with the preference vector **r** as loss weights, i.e the loss for input **r** is $\sum_i r_i \ell_i$
- PHN-EPO treats the preference **r** as a ray in loss space and trains $\theta(\psi, \mathbf{r})$ to reach a Pareto optimal point on the inverse ray $\mathbf{r}^{-1}$, namely, $r_1 \cdot \ell_1 = \cdots = r_m \cdot \ell_m$.

An Illustrative example

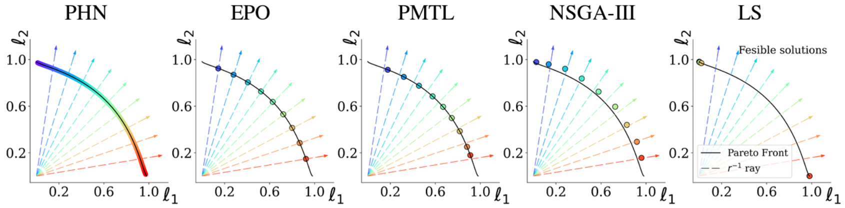
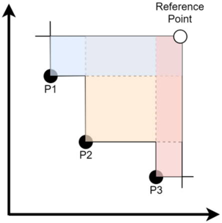


Figure 1: Illustrative example using the popular task of Fonseca (1995): demonstrating the relation between Pareto front, preference rays, and solutions. Pareto front (black solid line) for a 2D loss space and several rays (colored dashed lines) which represent various possible preferences.

Evaluation metrics

- **Hypervolume metric**: Given a set of points $S \subset \mathbb{R}^n$ and a reference point $\boldsymbol{\rho} \in \mathbb{R}^n_+$, the hypervolume of S is measured by the region of non-dominated points bounded above by $\boldsymbol{\rho}$.



- **Uniformity metric** quantifies how well the loss vector $\boldsymbol{\ell}(\theta)$ is aligned with the input ray r.

$$\mu_{\mathbf{r}}(\boldsymbol{\ell}(\theta)) = D_{KL}(\hat{\boldsymbol{\ell}}||\mathbf{1}/m), \text{ where } \hat{\boldsymbol{\ell}}_j = \frac{r_j \ell_j}{\sum_i r_i \ell_i}$$

Experiments

Multitask classification

	Multi-Fashion+MNIST		Multi-Fashion		Multi-MNIST		
	HV ⇑	Unif. ⇑	HV ⇑	Unif. ⇑	HV ⇑	Unif. ⇑	Run-time (min., Tesla V100)
LS	2.70	0.849	2.14	0.835	2.85	0.846	$9.0 \times 5 = 45$
CPMTL	2.76	-	2.16	-	2.88	-	$10.2 \times 5 = 51$
PMTL	2.67	0.776	2.13	0.192	2.86	0.793	$17.0 \times 5 = 85$
EPO	2.67	0.892	2.15	0.906	2.85	0.918	$23.6 \times 5 = 118$
PHN-LS (ours)	2.75	0.894	**2.19**	0.900	**2.90**	0.901	**12**
PHN-EPO (ours)	**2.78**	**0.952**	**2.19**	**0.921**	2.78	**0.920**	27

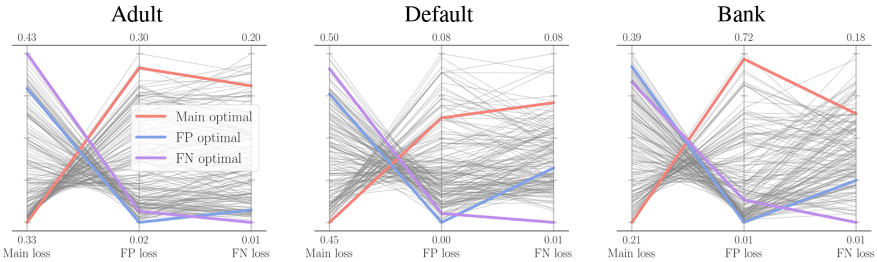
Semantic segmentation and Depth

	NYUv2		
	HV ⇑	Unif. ⇑	Run-time (hours, Tesla V100)
LS	3.550	0.666	$0.58 \times 5 = 2.92$
PMTL	3.554	0.679	$0.96 \times 5 = 4.79$
CPMTL	3.570	-	$0.71 \times 5 = 3.55$
EPO	3.266	0.728	$1.02 \times 5 = 5.11$
PHN-LS (ours)	3.546	0.798	**0.67**
PHN-EPO (ours)	**3.589**	**0.820**	1.04

Applications

Fairness

A 3-dimensional optimization problem, with a classification objective and two fairness objectives: False Positive (FP) fairness, and False Negative (FN) fairness.



Literature

[1] Aviv Navon et al. *Learning the Pareto Front with Hypernetworks*. 2020. DOI: 10.48550/ARXIV.2010.04104. URL: https://arxiv.org/abs/2010.04104.